Recent Advances in Integrating

Machine Learning and Combinatorial Optimization

Elias B. Khalil

Andrea Lodi

Didier Chételat

Antoine Prouvost

35th AAAI Conference

Bistra Dilkina

Maxime Gasse

Giulia Zarpellon

Laurent Charlin

CANADA EXCELLENCE RESEARCH CHAIR
DATA SCIENCE FOR REAL-TIME DECISION-MAKING
UNIVERSITY OF TORONTO
POLYTECHNIQUE MONTRÉAL
USC University of Southern California

Part 1

Introduction to Combinatorial Optimization

Elias Khalil, Andrea Lodi, Bistra Dilkina, Didier Chételat,
Maxime Gasse, Antoine Prouvost, Giulia Zarpellon, Laurent Charlin

MH7: Recent Advances in Integrating
Machine Learning and Combinatorial Optimization

35th AAAI Conference
on Artificial Intelligence
February 3rd 2021

Humans learn to **design** **algorithms**.

Humans learn to **design** **algorithms**.

Can **algorithms** “learn” to **design** **algorithms**?

Humans learn to **design algorithms**.

Machine Learning

Can **algorithms** “learn” to **design algorithms**?

Humans learn to **design algorithms**.

Machine Learning

Can **algorithms** “learn” to **design algorithms**?

Combinatorial Optimization

Data Center Resource Management

Photo from: https://www.reit.com/what-reit/reit-sectors/data-center-reits

Data Center Resource Management

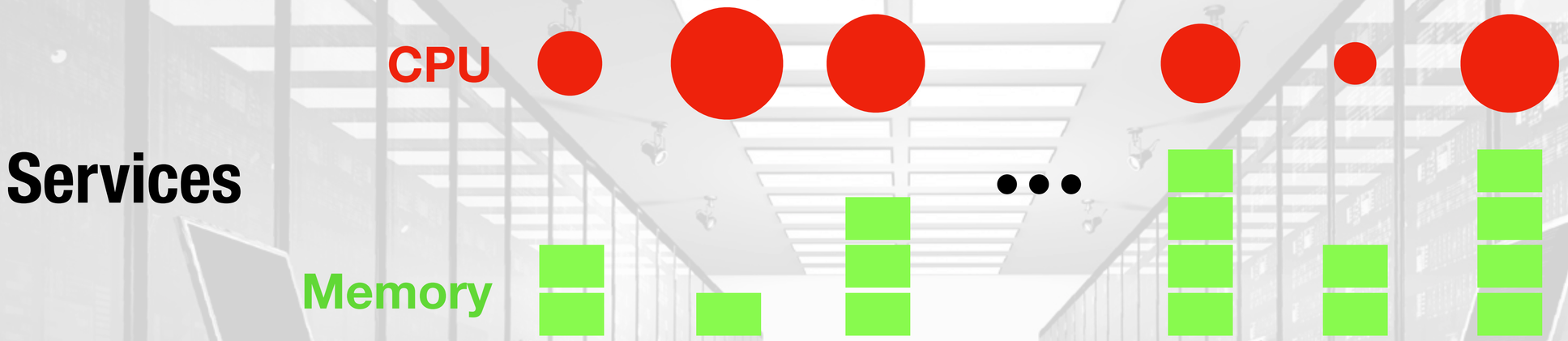


Photo from: https://www.reit.com/what-reit/reit-sectors/data-center-reits

Data Center Resource Management

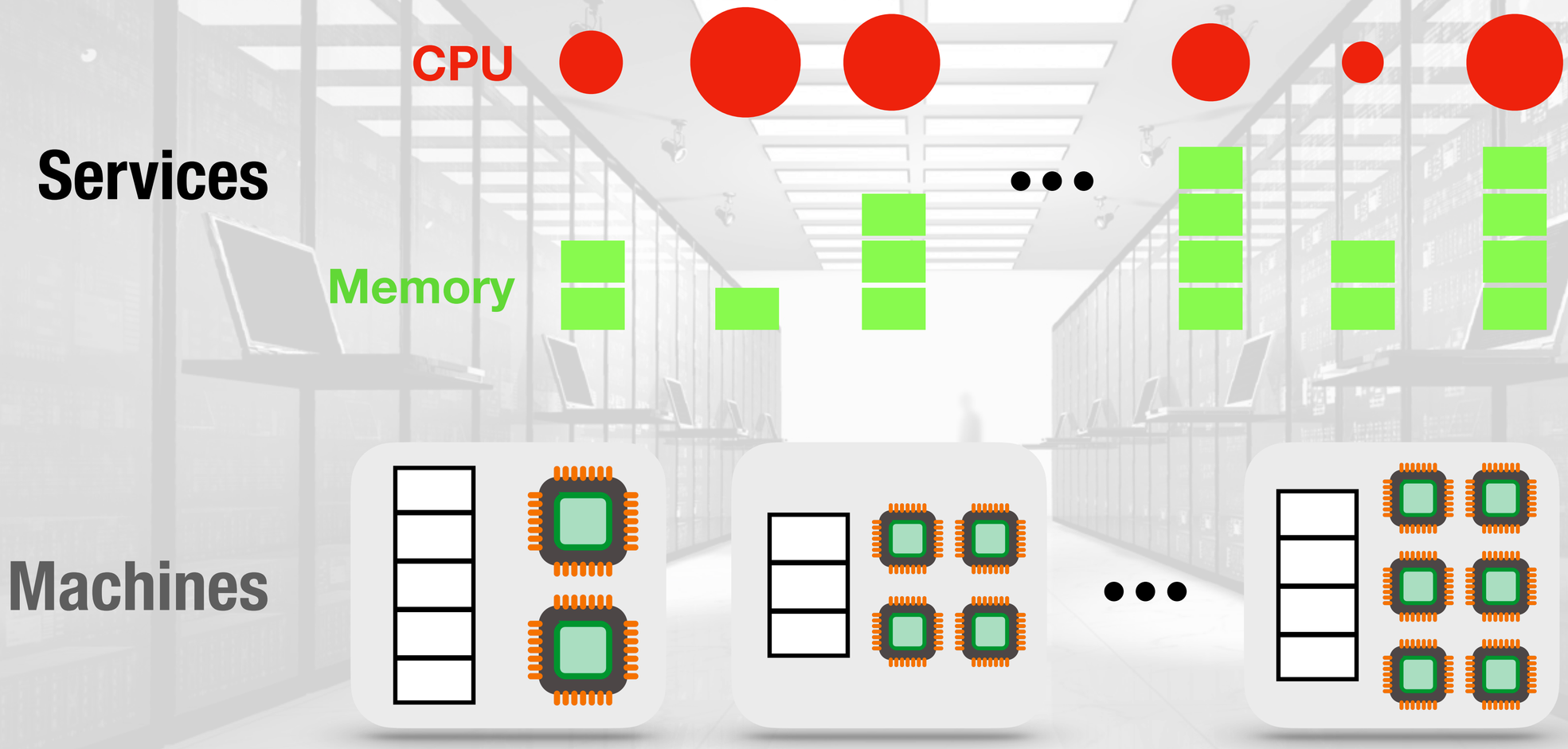


Photo from: https://www.reit.com/what-reit/reit-sectors/data-center-reits

Data Center Resource Management

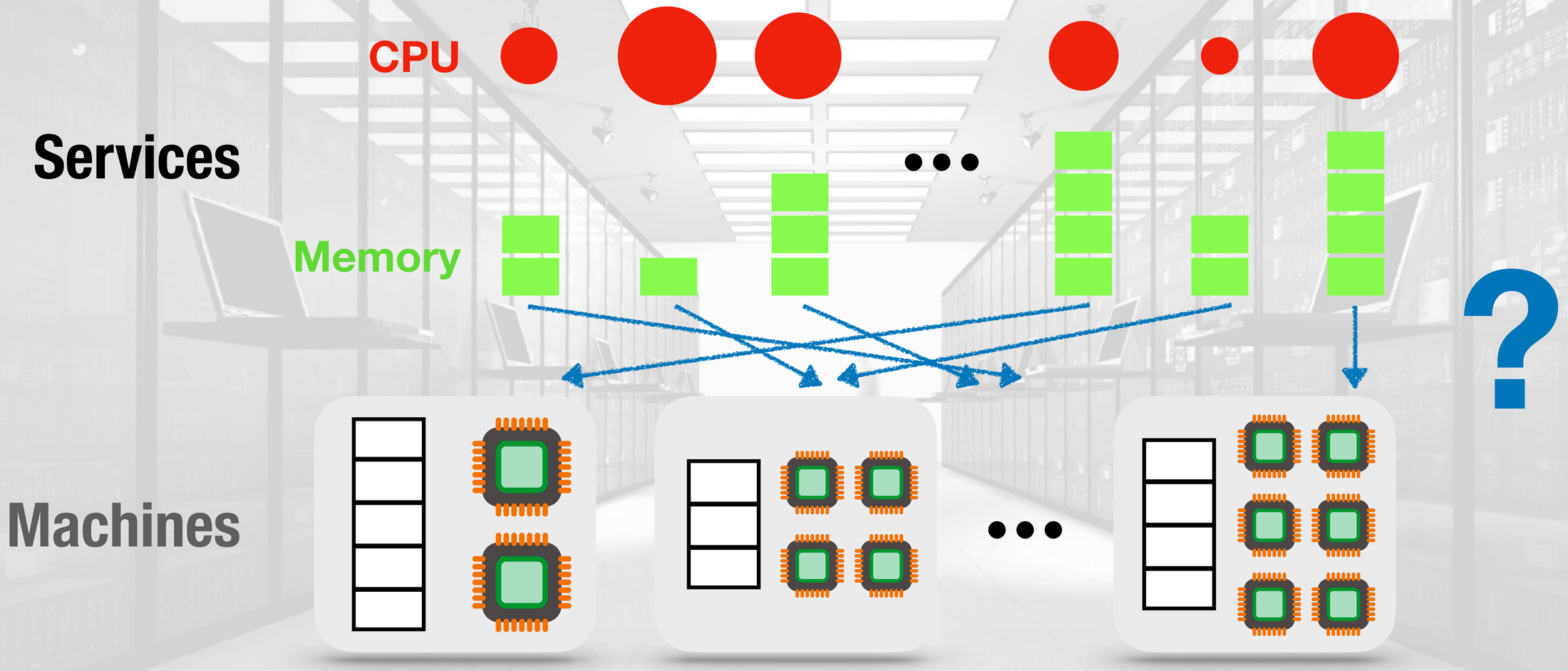
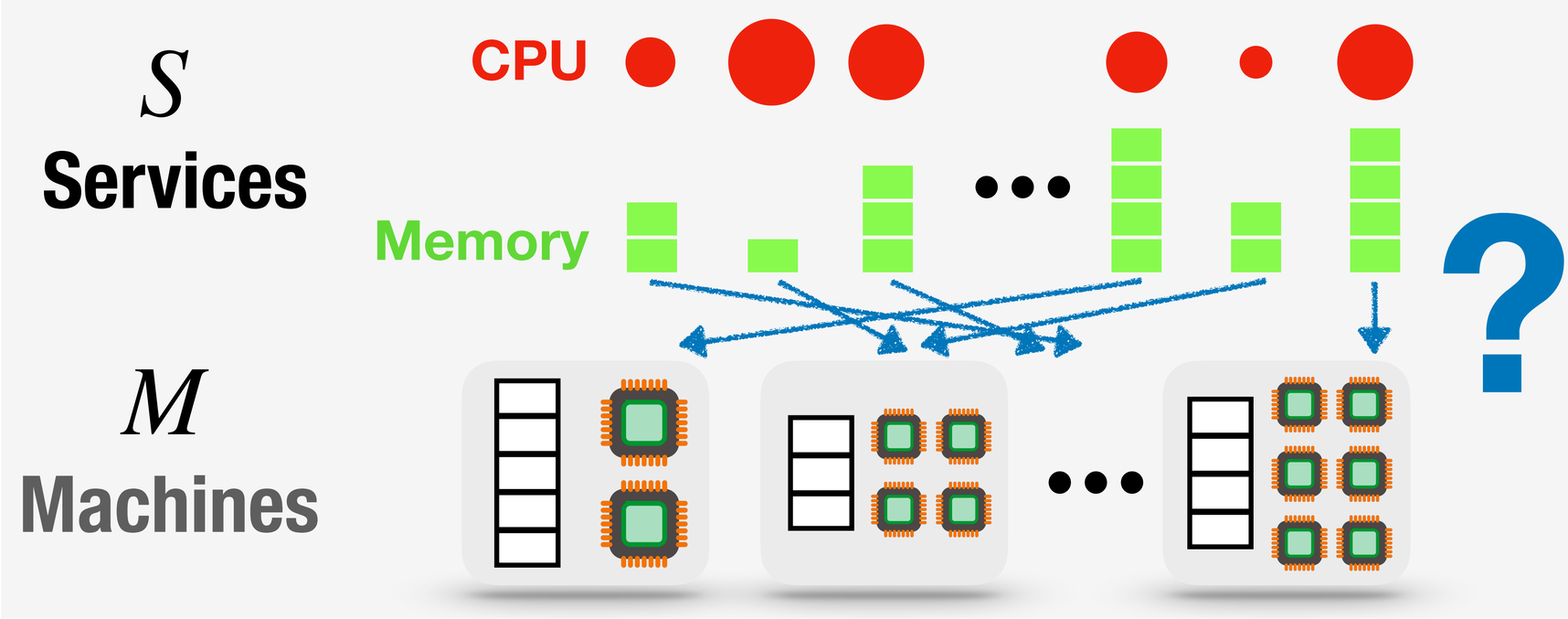
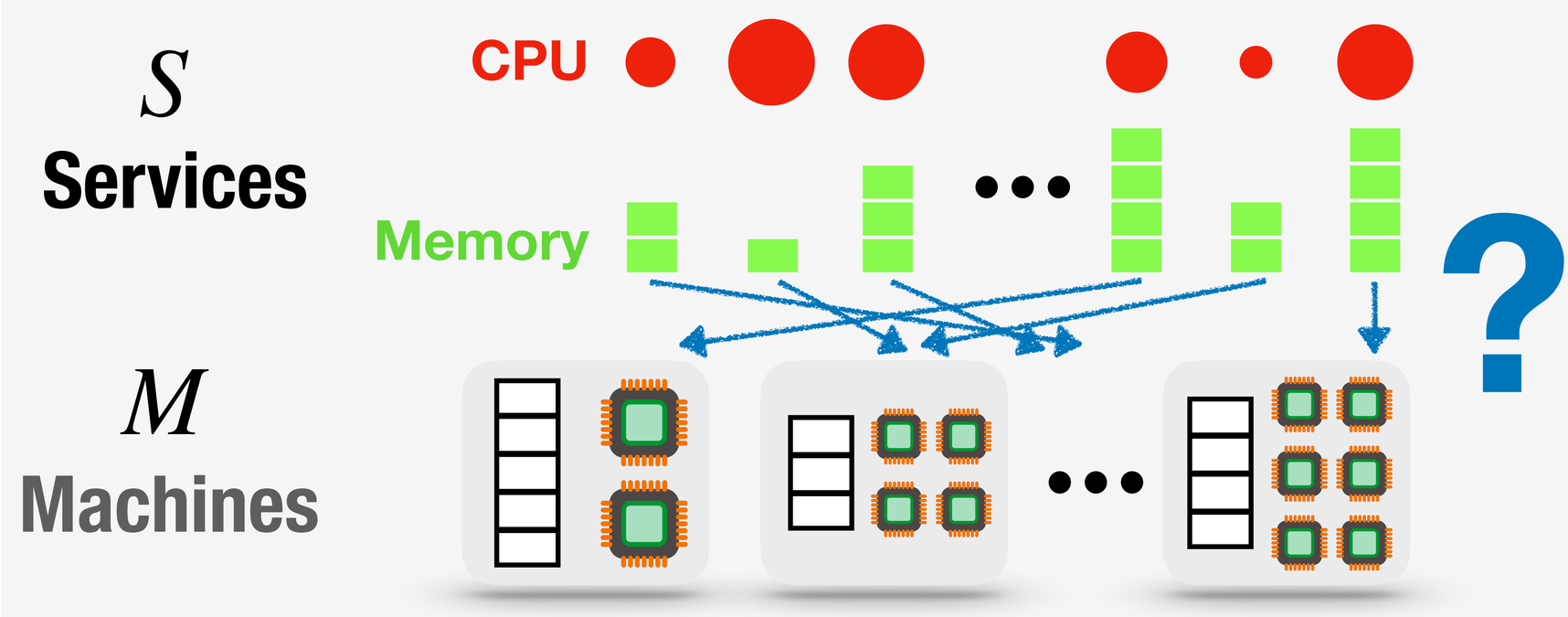


Photo from: https://www.reit.com/what-reit/reit-sectors/data-center-reits


S
Services
CPU
Memory
M
Machines
?

$y_m = 1$ if machine m is used

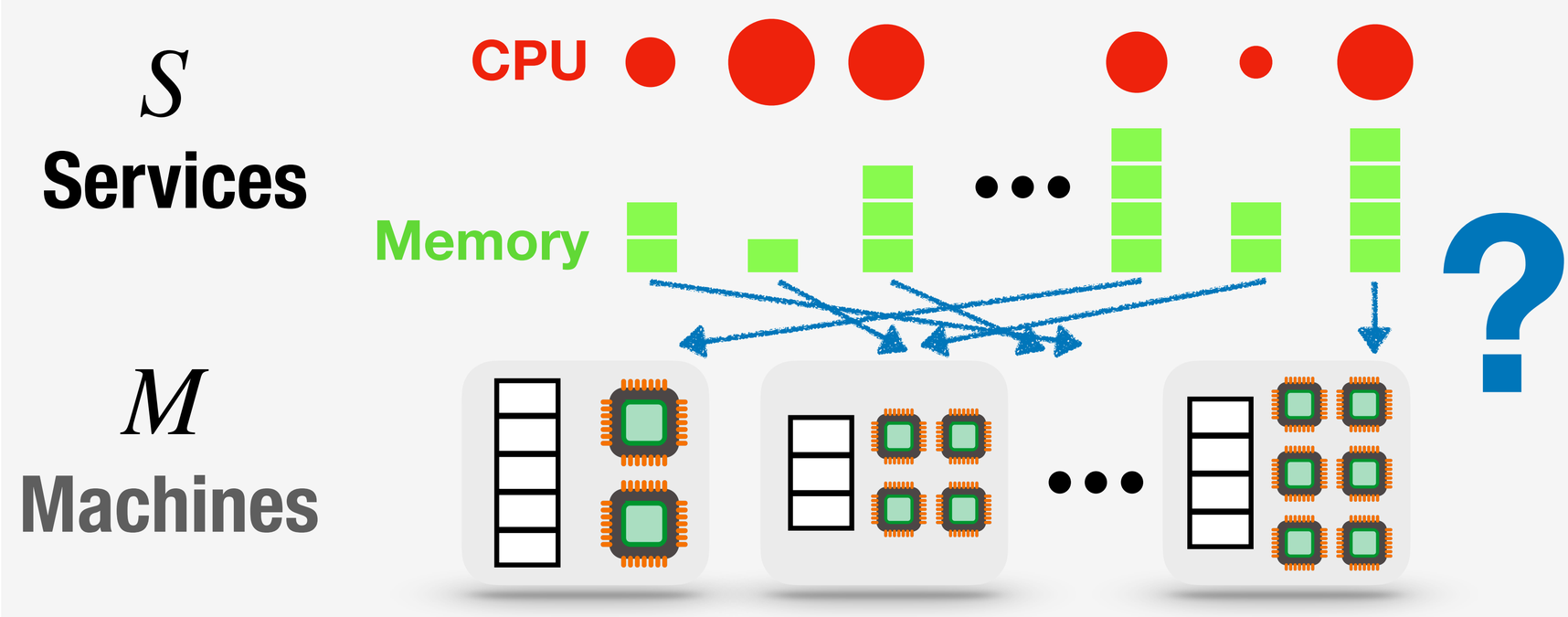
$x_{s,m} = 1$ if service s runs on m



$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

$x \in \{0,1\}^{S \times M}, y \in \{0,1\}^M$

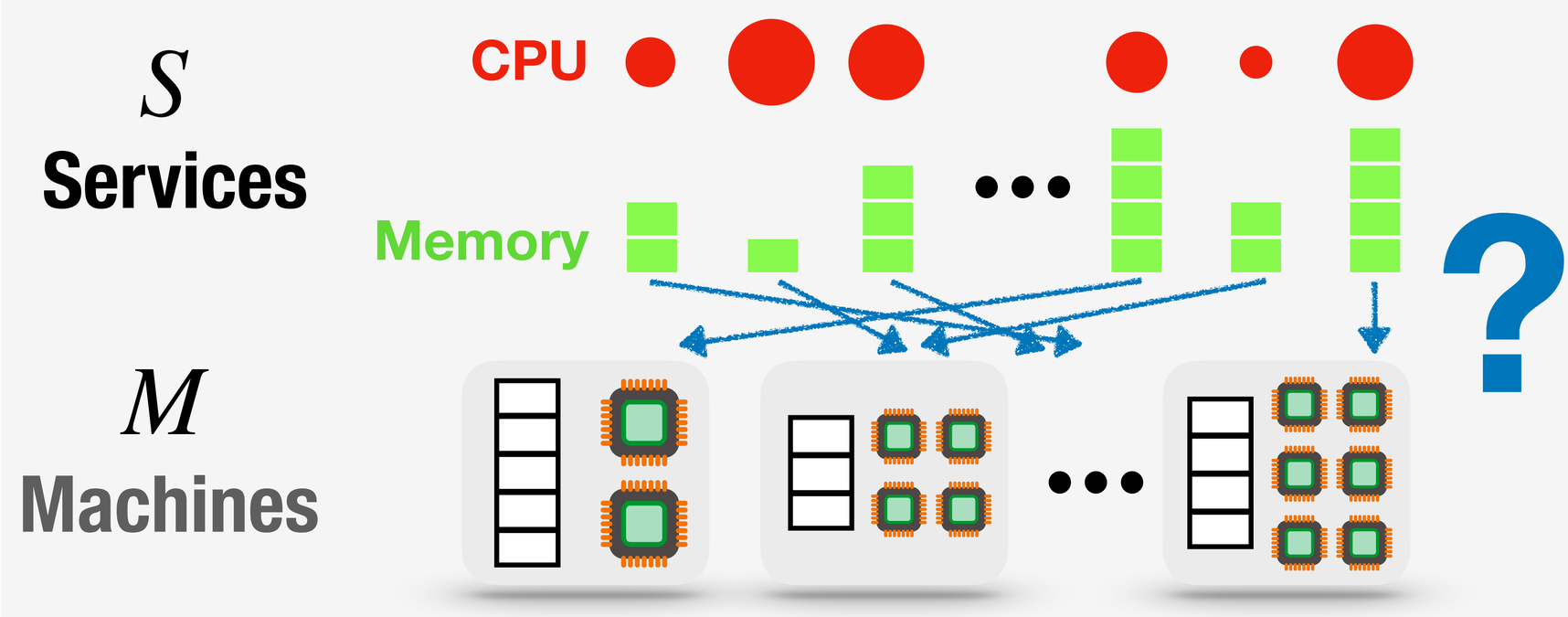


$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

$$x \in \{0,1\}^{S \times M},\ y \in \{0,1\}^{M}$$

$$\textbf{minimize} \sum_{m=1}^{M} y_m$$



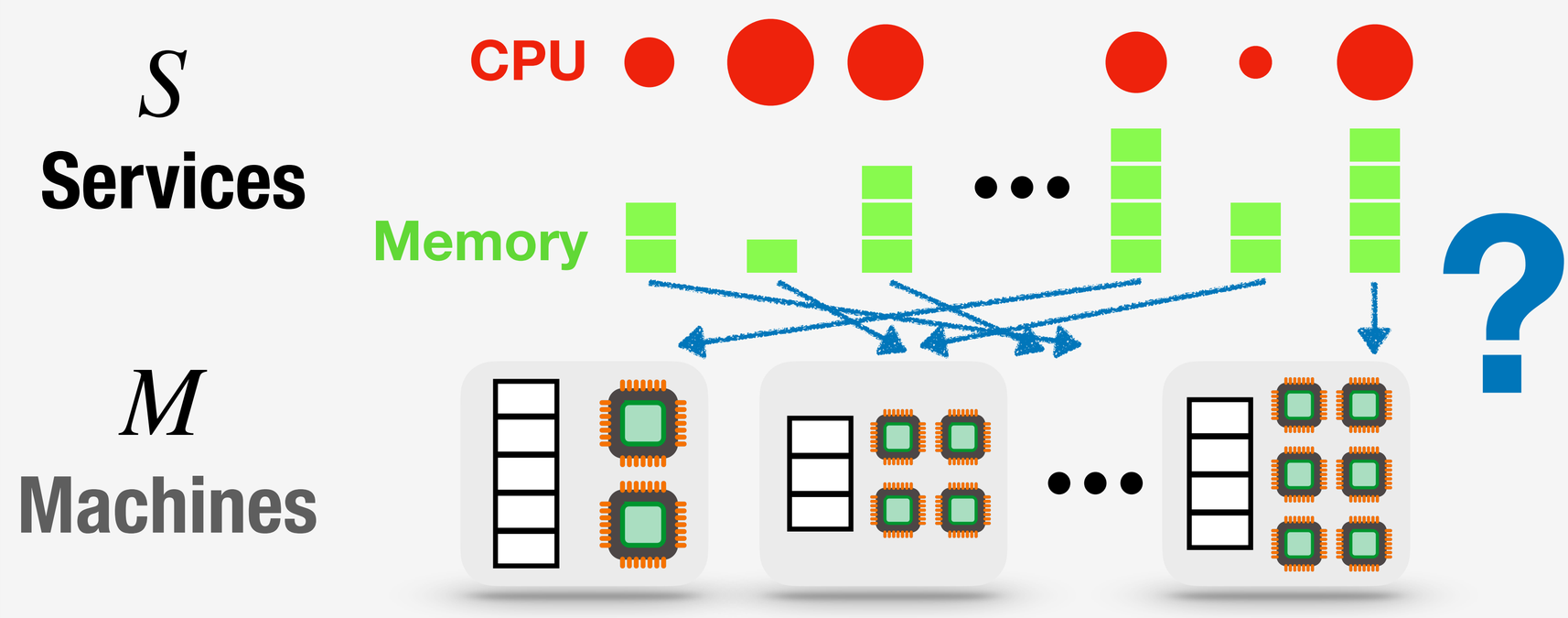
$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

$$x \in \{0,1\}^{S \times M},\ y \in \{0,1\}^{M}$$

$$\textbf{minimize} \sum_{m=1}^{M} y_m$$

Constraints:



$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

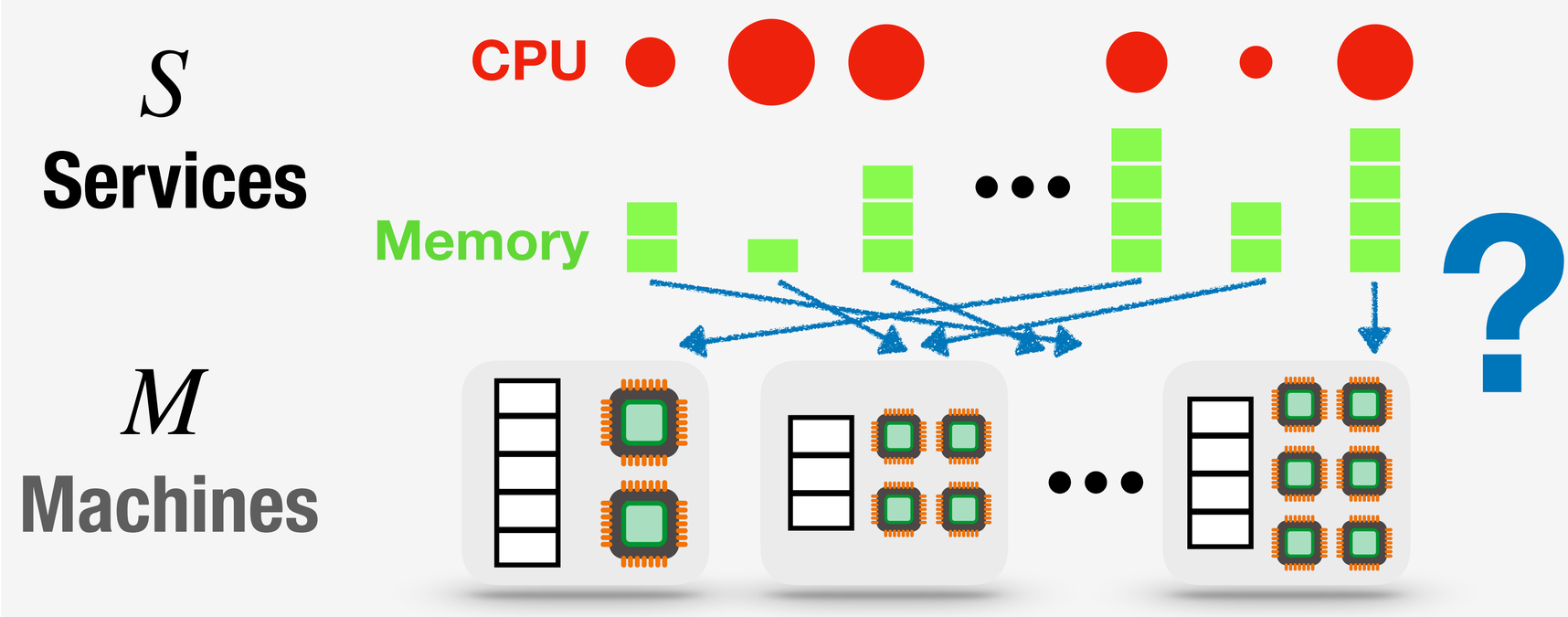
$x \in \{0,1\}^{S \times M}, y \in \{0,1\}^M$

minimize $\sum_{m=1}^{M} y_m$

Constraints:

Each service on one machine only

$$\sum_{m=1}^{M} x_{s,m} = 1 \quad \forall s$$

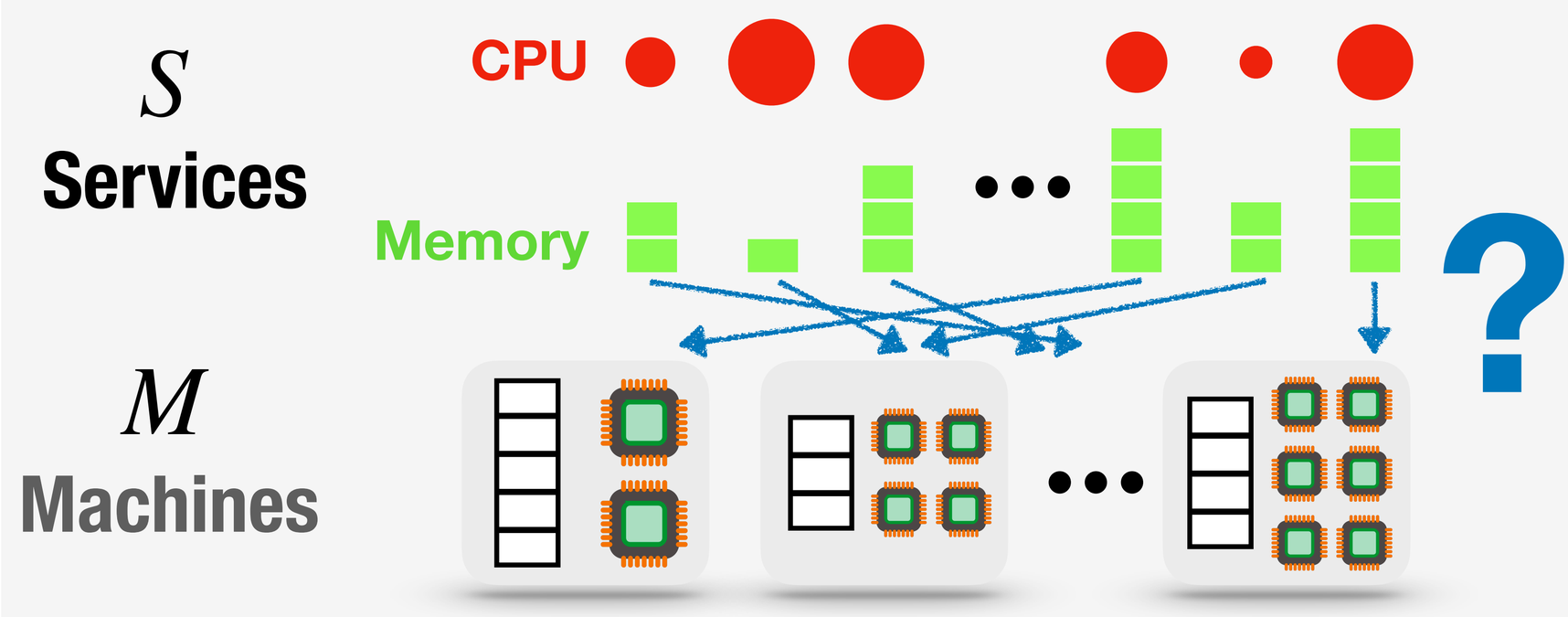


$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

$x \in \{0,1\}^{S \times M}, y \in \{0,1\}^M$

minimize $\sum_{m=1}^{M} y_m$



Constraints:

Each service on one machine only

$$\sum_{m=1}^{M} x_{s,m} = 1 \quad \forall s$$

$$y_m \geq x_{s,m} \quad \forall s, m$$

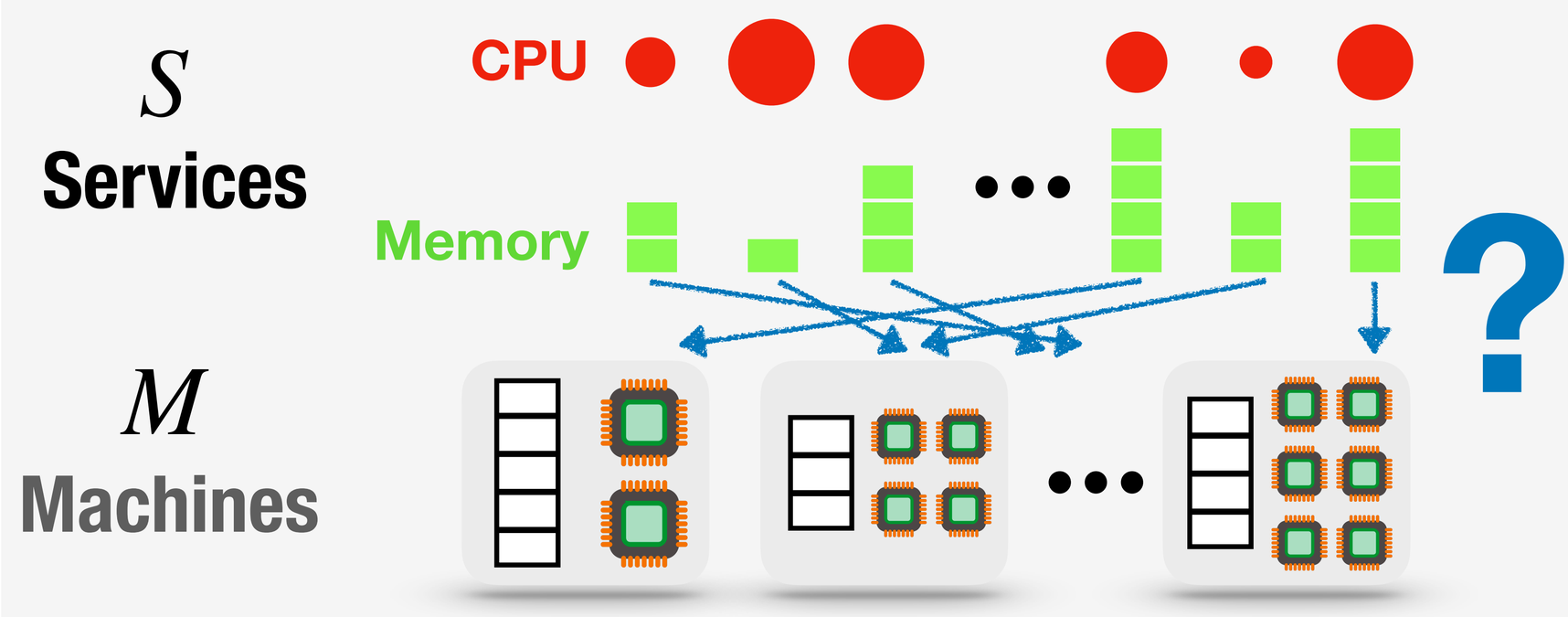
Machine is "ON" if a job is assigned to it

$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

$x \in \{0,1\}^{S \times M}, y \in \{0,1\}^M$

$$\textbf{minimize} \sum_{m=1}^{M} y_m$$



Constraints:

Each service on one machine only

$$\sum_{m=1}^{M} x_{s,m} = 1 \quad \forall s$$

Memory capacity

$$\sum_{s=1}^{S} \textbf{mem}(s) \cdot x_{s,m} \leq \textbf{cap-mem}(m) \quad \forall m$$

$$y_m \geq x_{s,m} \quad \forall s, m$$

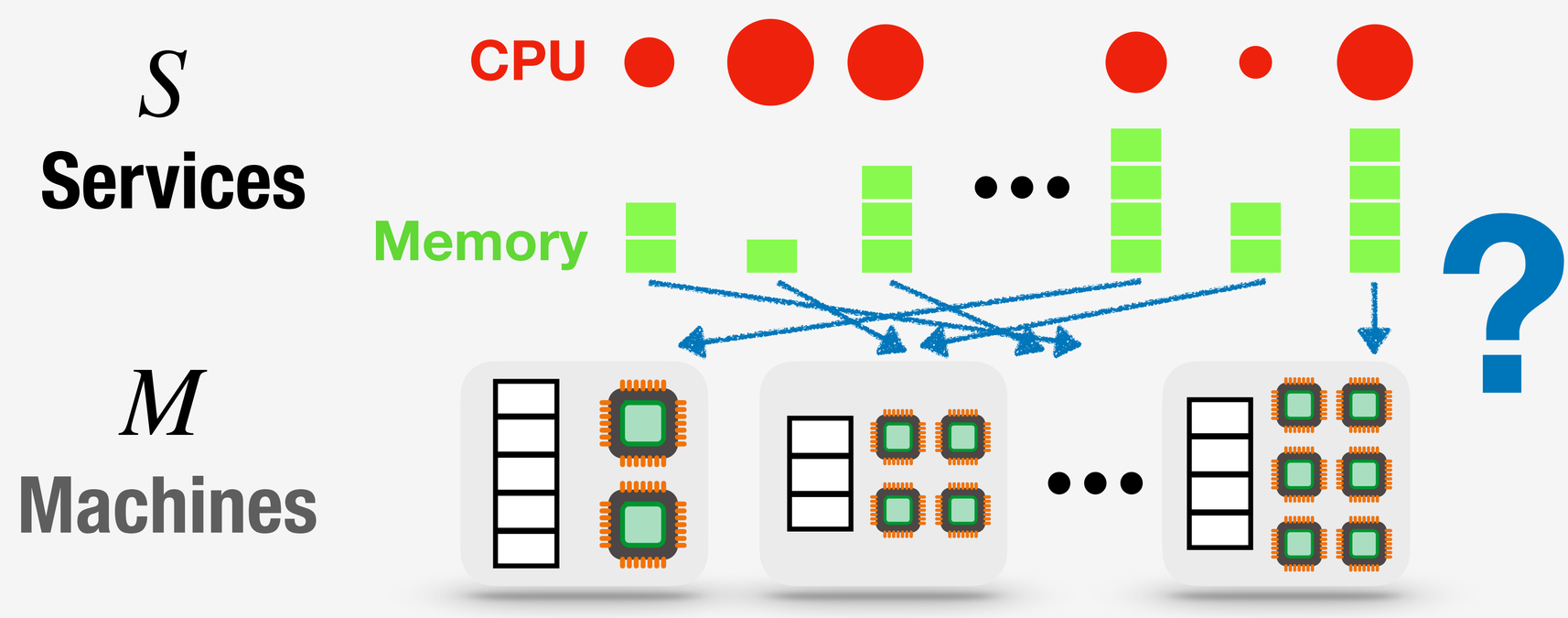
Machine is "ON" if a job is assigned to it

$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

$x \in \{0,1\}^{S \times M}, y \in \{0,1\}^M$

$$\textbf{minimize} \sum_{m=1}^{M} y_m$$



Constraints:

Each service on one machine only

$$\sum_{m=1}^{M} x_{s,m} = 1 \quad \forall s$$

$$y_m \geq x_{s,m} \quad \forall s, m$$

Machine is "ON" if a job is assigned to it

Memory capacity

$$\sum_{s=1}^{S} \textbf{mem}(s) \cdot x_{s,m} \leq \textbf{cap-mem}(m) \quad \forall m$$

$$\sum_{s=1}^{S} \textbf{cpu}(s) \cdot x_{s,m} \leq \textbf{cap-cpu}(m) \quad \forall m$$

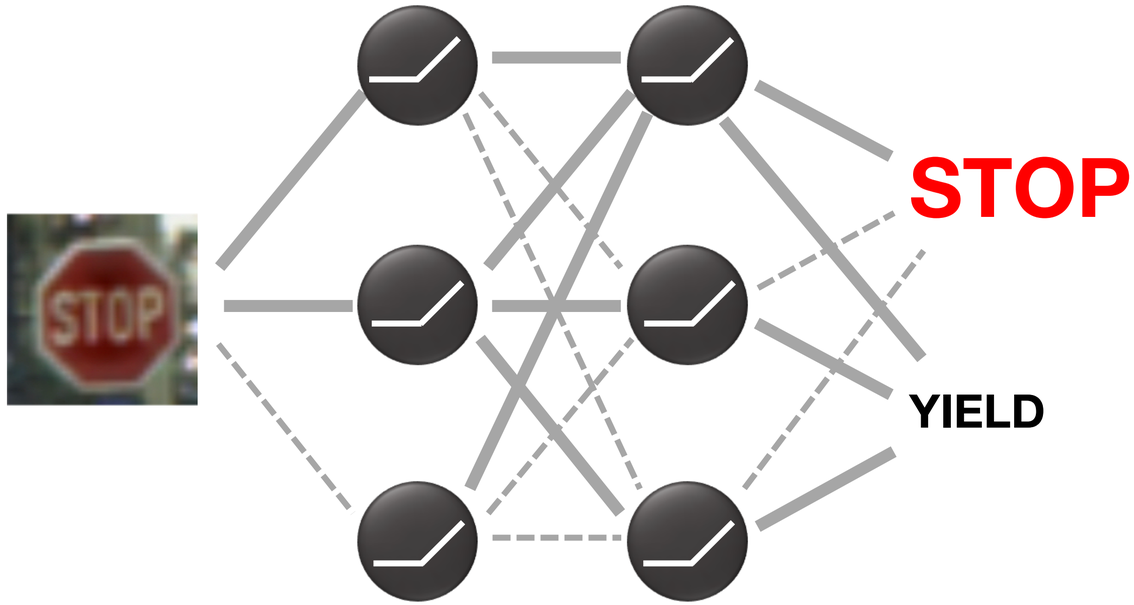
Processor capacity

Safety-Critical Machine Learning

See AAAI-21 Tutorial “AH3: From Explainability to Model Quality and Back Again”

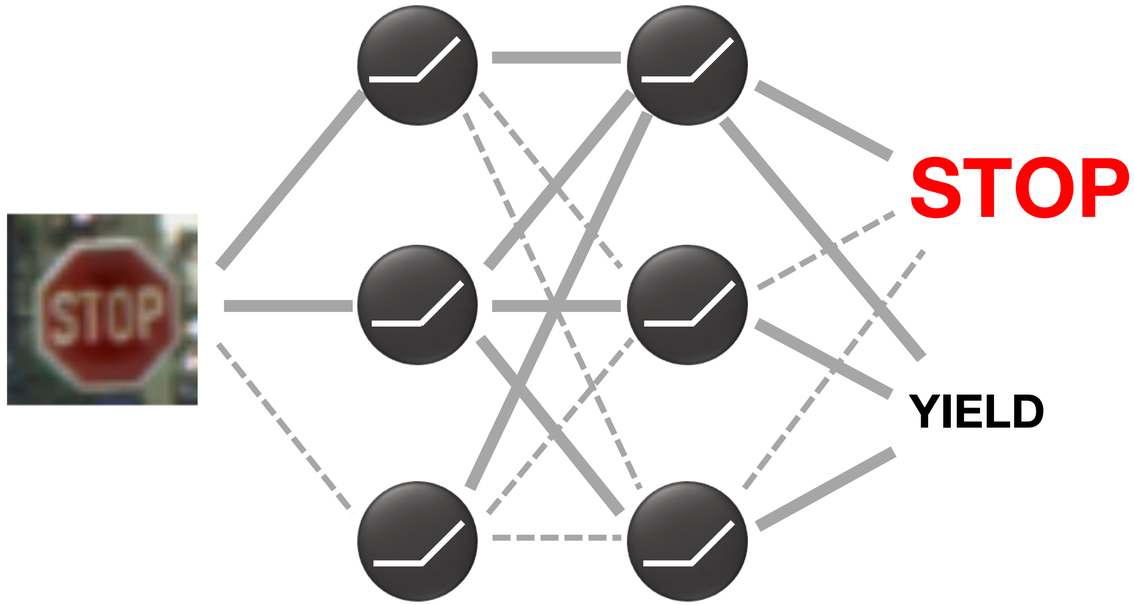
Safety-Critical Machine Learning

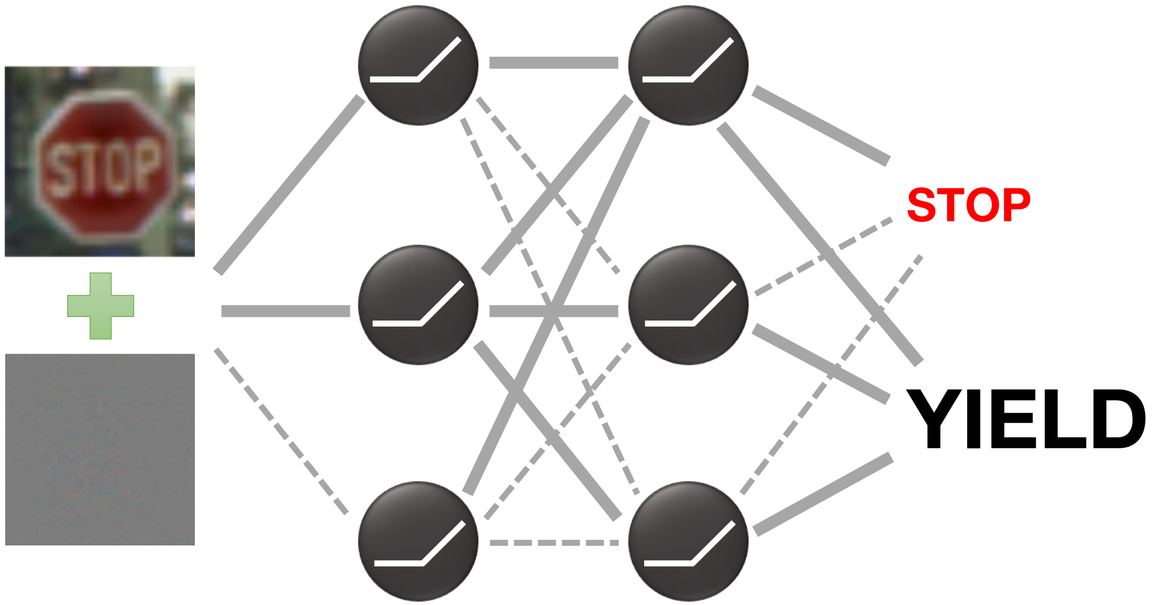
See AAAI-21 Tutorial “AH3: From Explainability to Model Quality and Back Again”



Safety-Critical Machine Learning

See AAAI-21 Tutorial “AH3: From Explainability to Model Quality and Back Again”

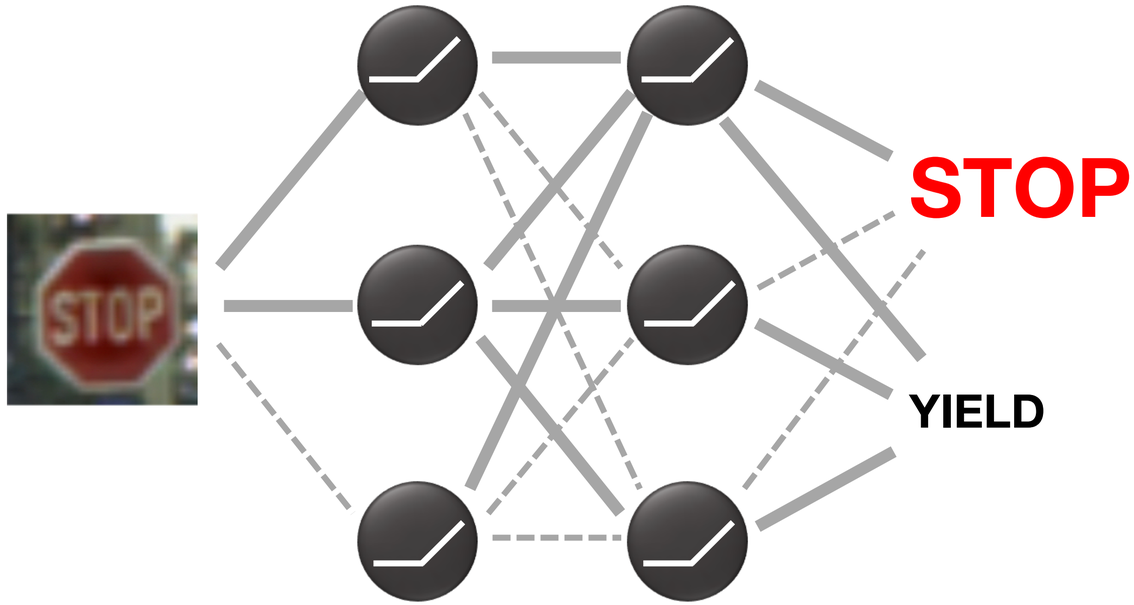


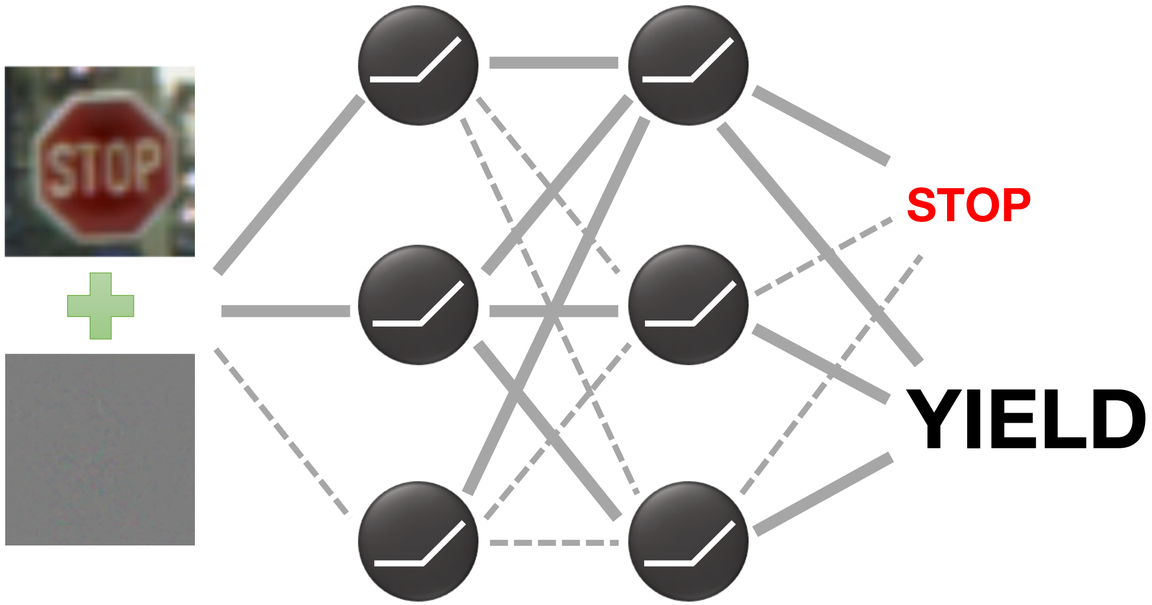


Safety-Critical Machine Learning

See AAAI-21 Tutorial "AH3: From Explainability to Model Quality and Back Again"

Goal: **Guarantee** that trained model has **desirable behavior**





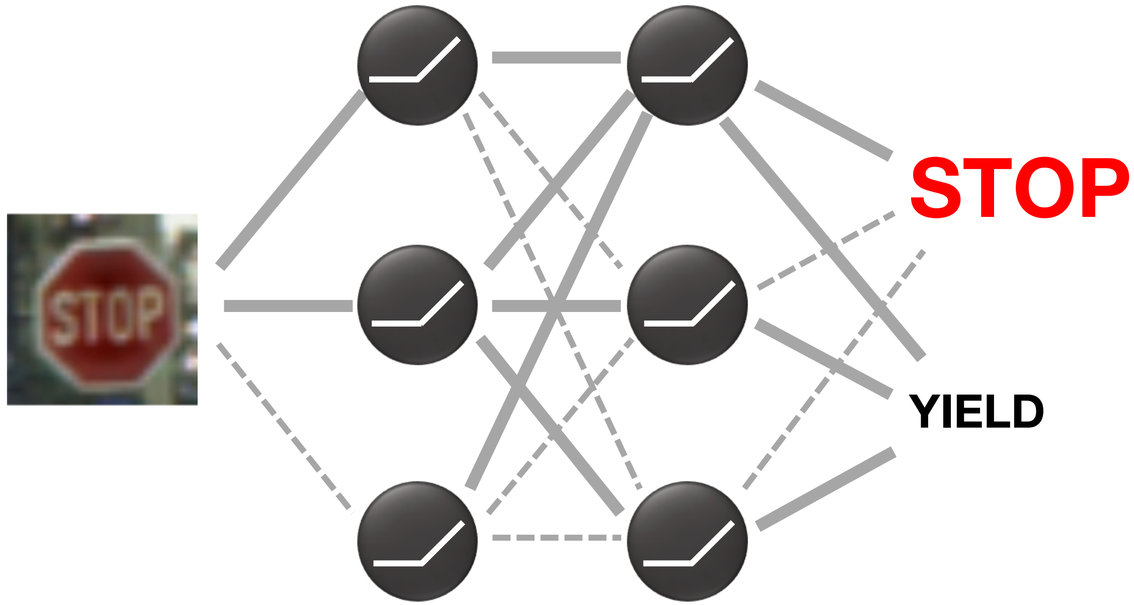
Safety-Critical Machine Learning

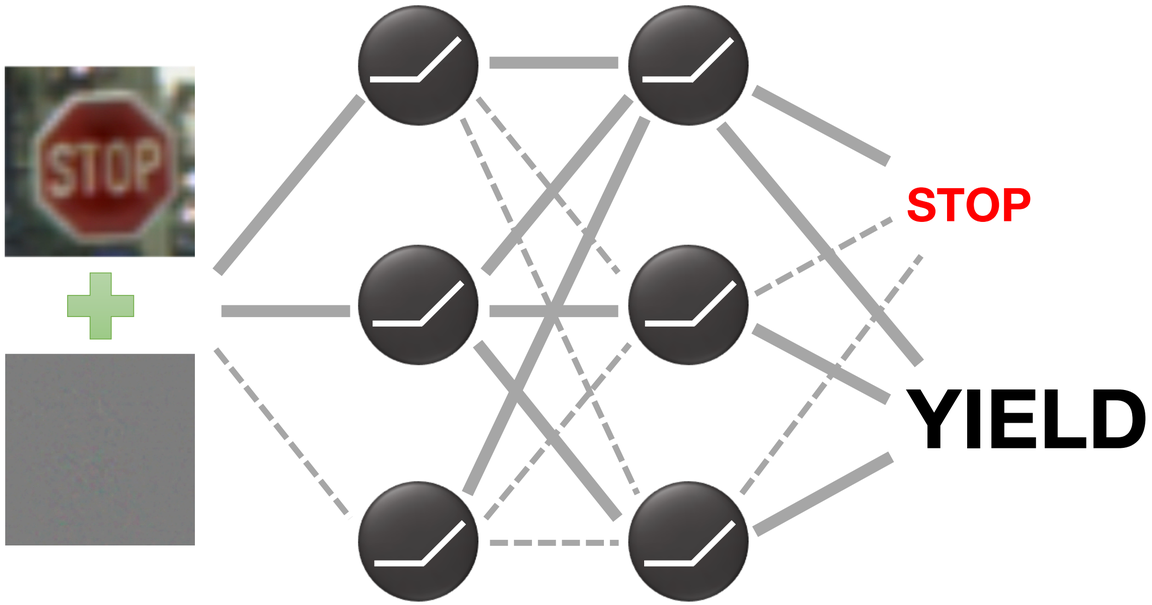
See AAAI-21 Tutorial "AH3: From Explainability to Model Quality and Back Again"

Goal: **Guarantee** that trained model has **desirable behavior**

→

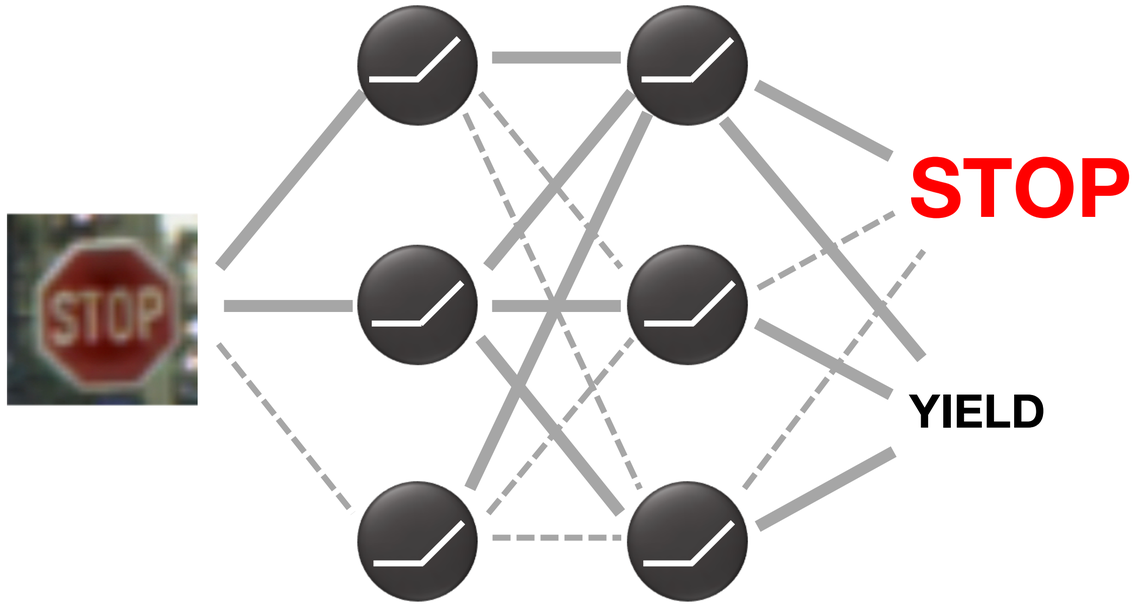
Model ReLU with **Binary** variables + Linear **Inequalities**

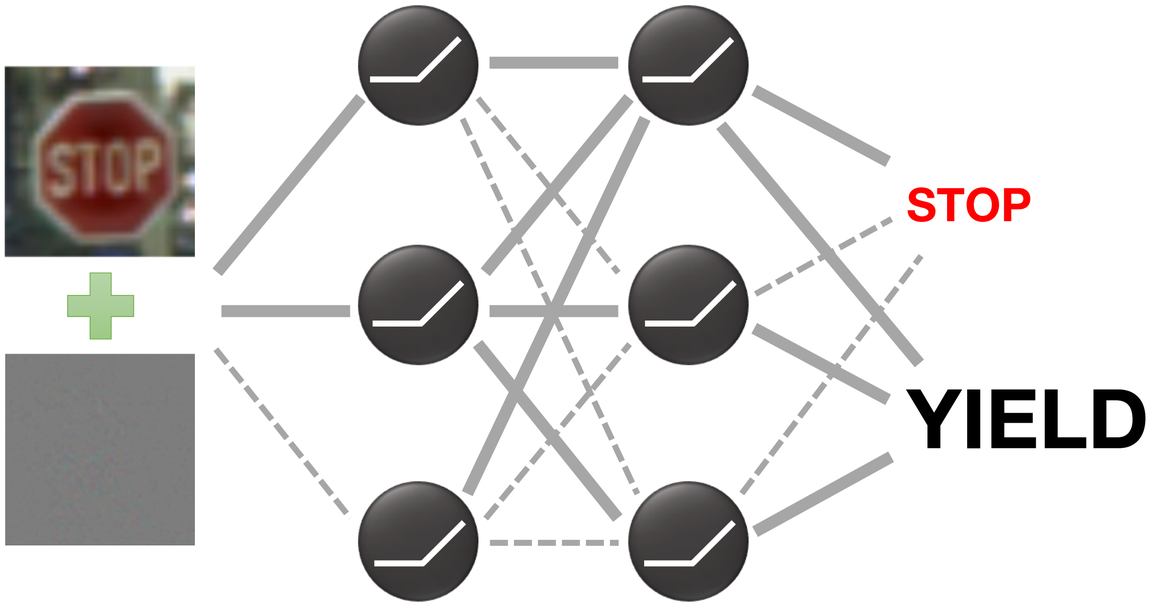




Safety-Critical Machine Learning

See AAAI-21 Tutorial “AH3: From Explainability to Model Quality and Back Again”





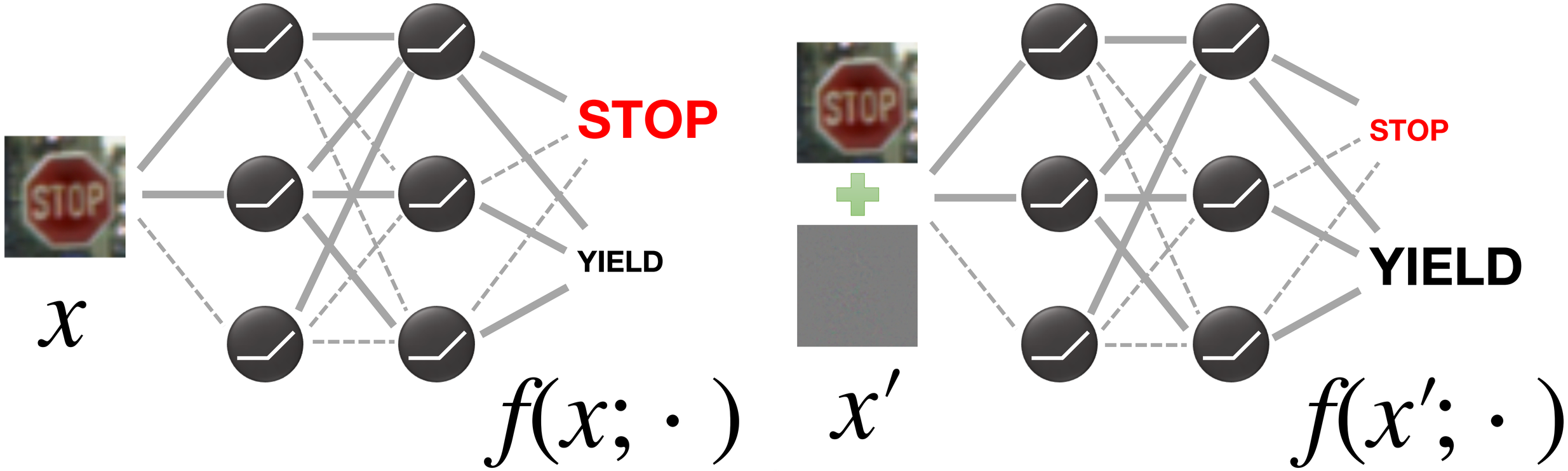
Safety-Critical Machine Learning

See AAAI-21 Tutorial "AH3: From Explainability to Model Quality and Back Again"

Verification Problem

prove $\nexists x'$ **close to** x

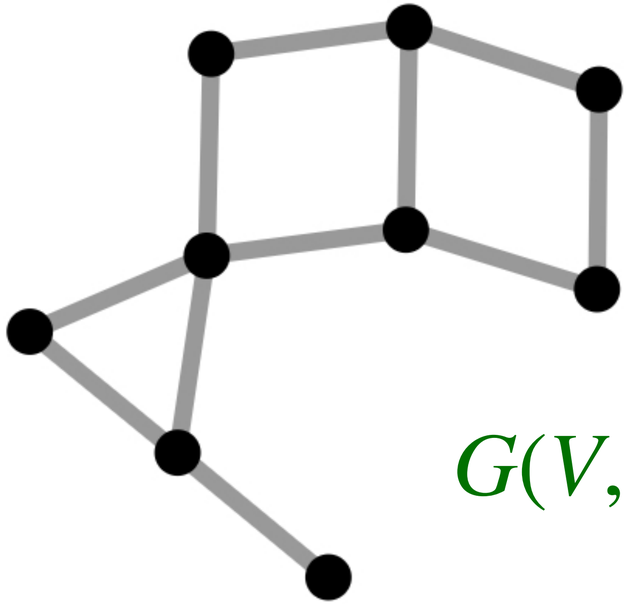
such that $f(x';$ **STOP**$) < f(x';$ **YIELD**$)$



Forest Harvesting

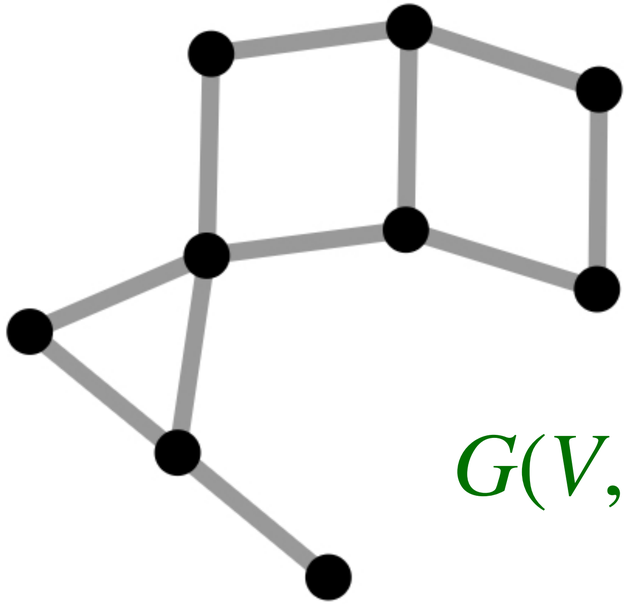
Forest Harvesting

Forest Harvesting



$G(V, E)$

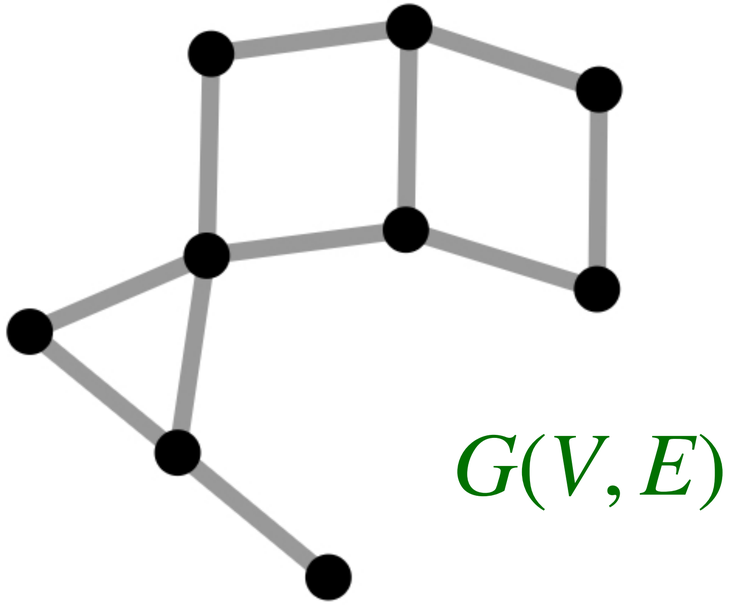
Forest Harvesting



$G(V, E)$

Forest Harvesting

Goal: Harvest subset of parcels to maximize **revenue**; pay **cost** for harvesting adjacent parcels



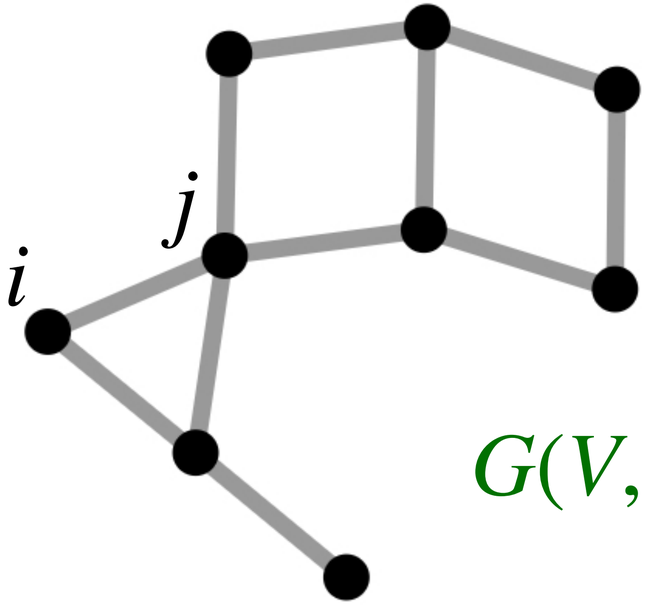
Forest Harvesting

Goal: Harvest subset of parcels to maximize **revenue**; pay **cost** for harvesting adjacent parcels

$$\textbf{maximize} \sum_{i \in V} r_i x_i - \sum_{(i,j) \in E} c_{ij} y_{ij}$$

$$\textbf{subject to } x_i + x_j - y_{ij} \leq 1$$

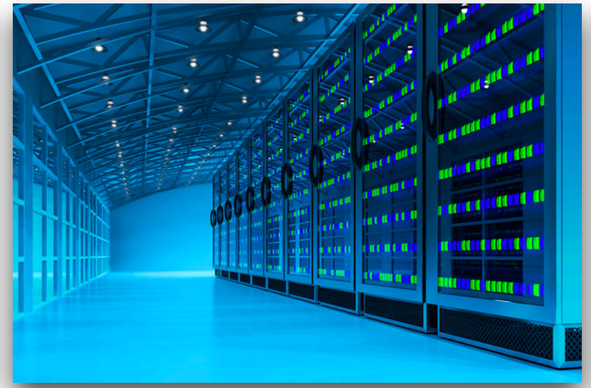
$$x \in \{0,1\}^n, y \in \{0,1\}^m$$



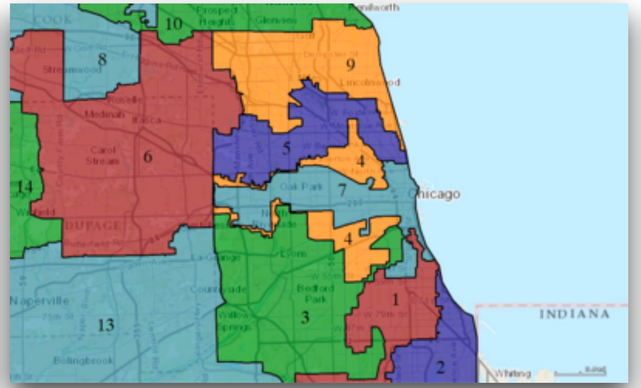
$G(V, E)$

Auction Design

Data Center Management



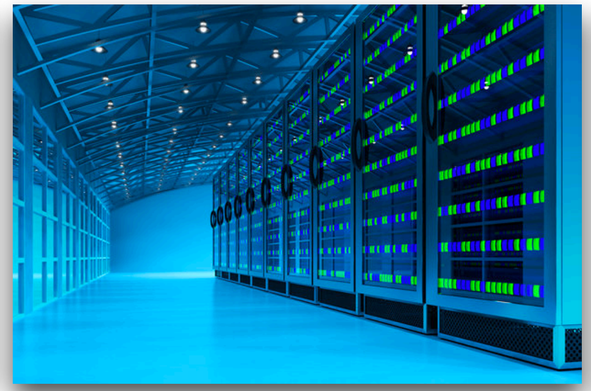
Political Districting



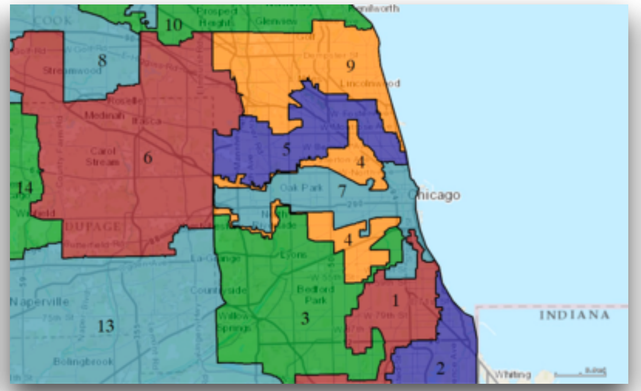
Kidney Exchange

Auction Design

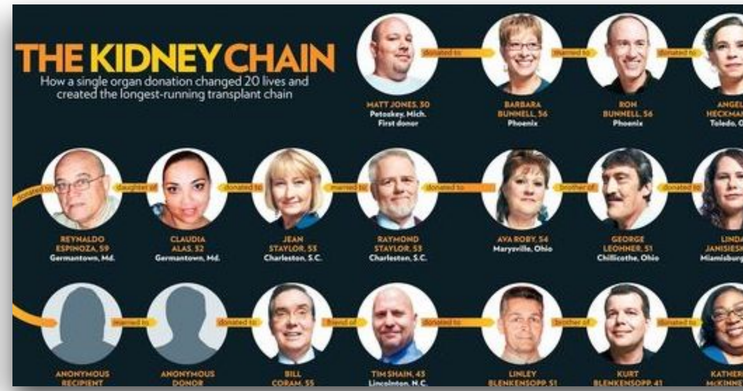
Data Center Management



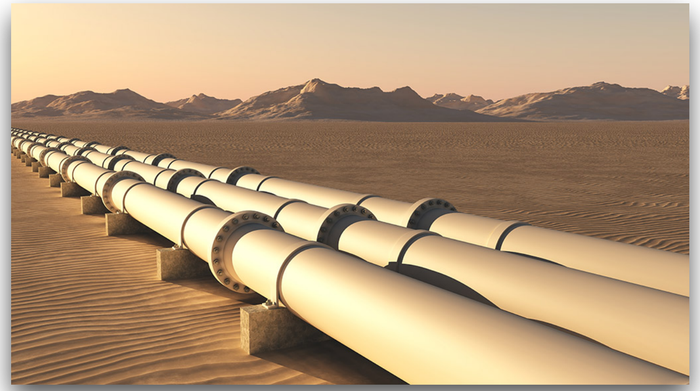
Political Districting



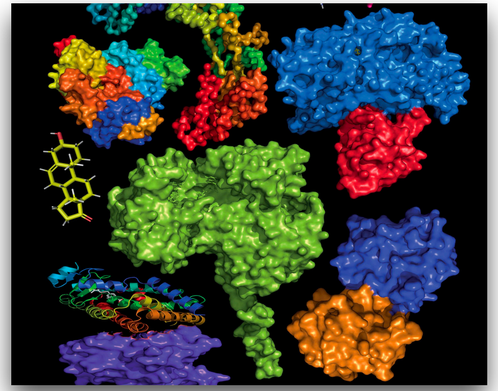
Kidney Exchange



Energy Systems



Scientific Discovery

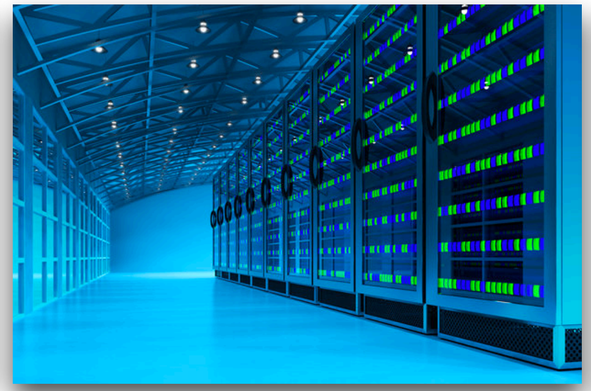


Ridesharing

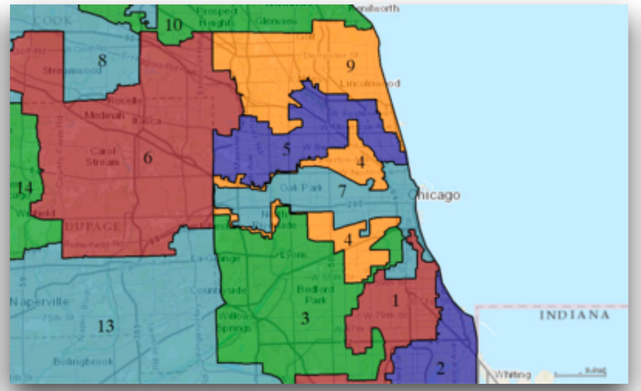
Cancer Therapeutics

Auction Design

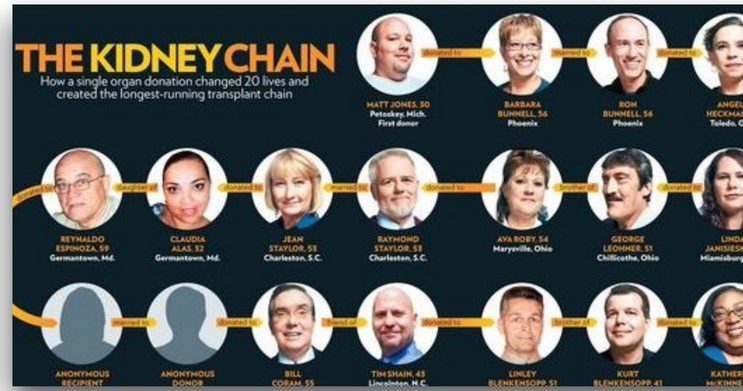
Data Center Management



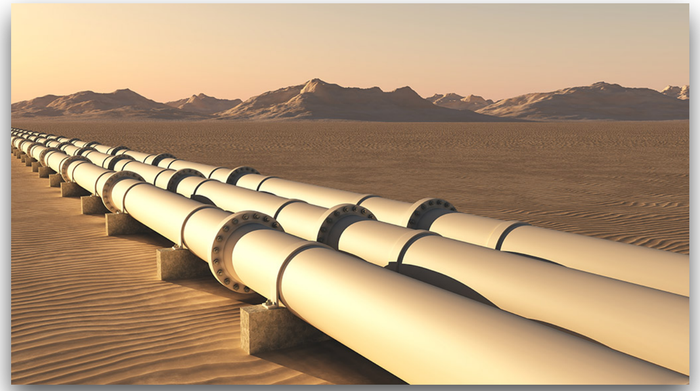
Political Districting



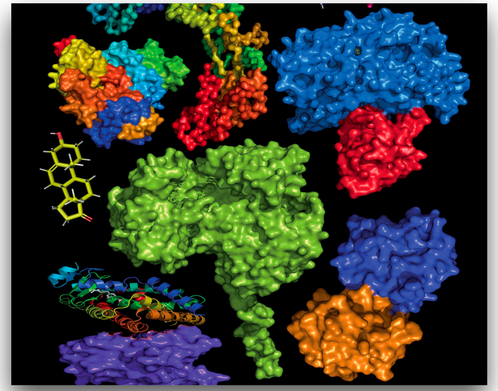
Kidney Exchange



Energy Systems



Scientific Discovery



Ridesharing

Cancer Therapeutics

Airline Scheduling

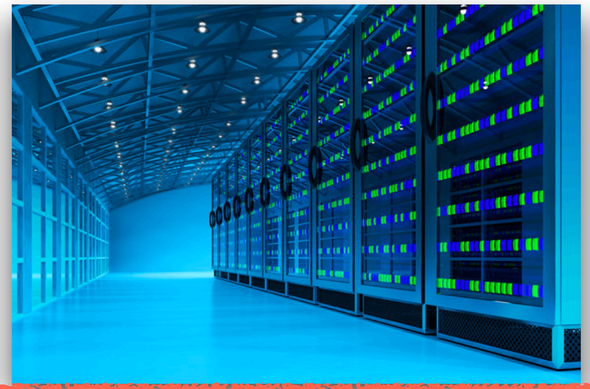
Conservation Planning

Disaster Response

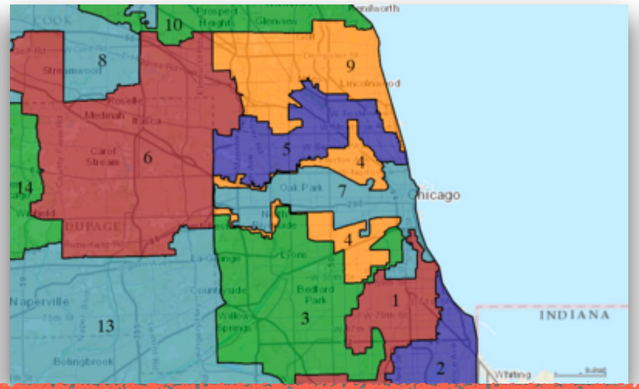
College Admissions

Auction Design

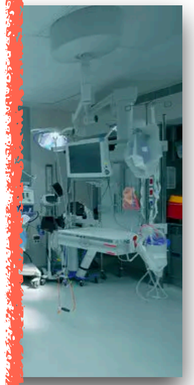
Data Center Management



Political Districting



Kidney Exchange



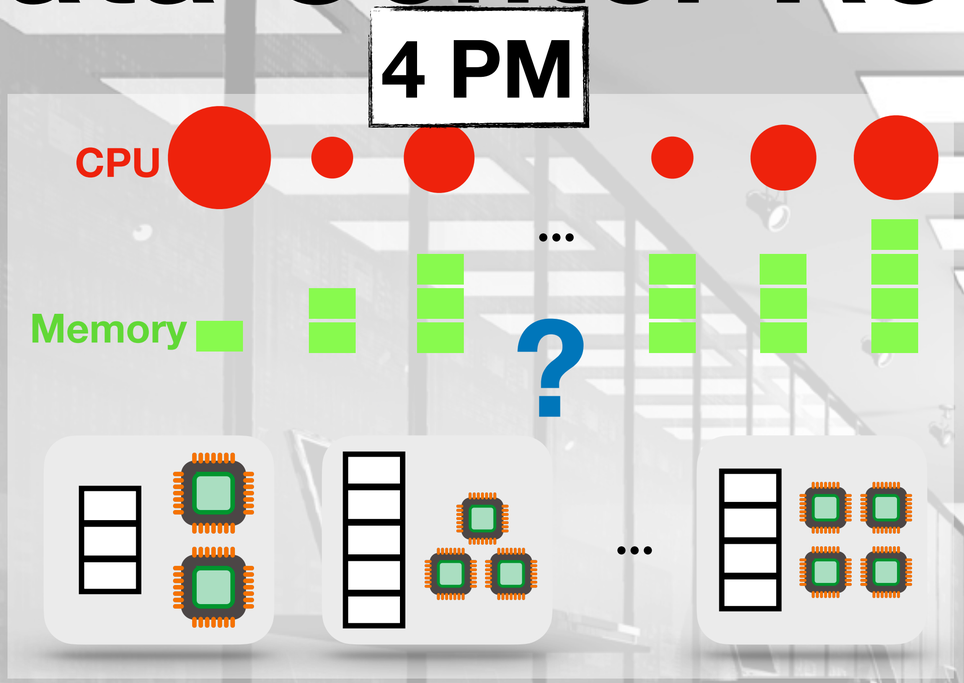
> 50% of INFORMS Edelman Award winners use Discrete Optimization

→ **Billions** ($) in savings/profit

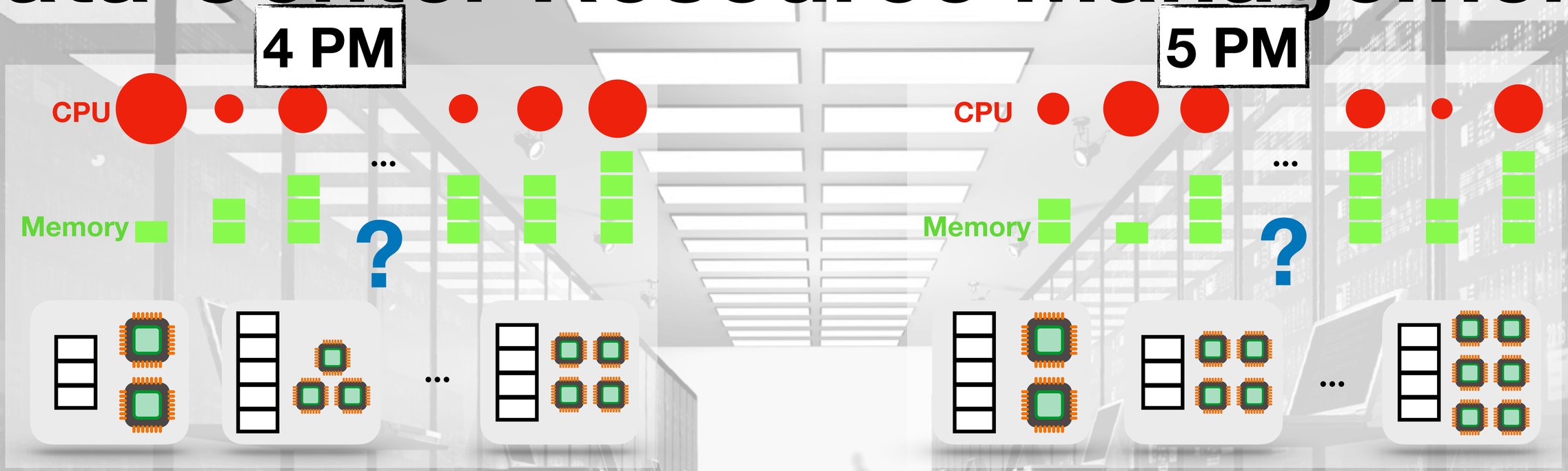
George Nemhauser, Plenary at EURO INFORMS, 2013

Data Center Resource Management

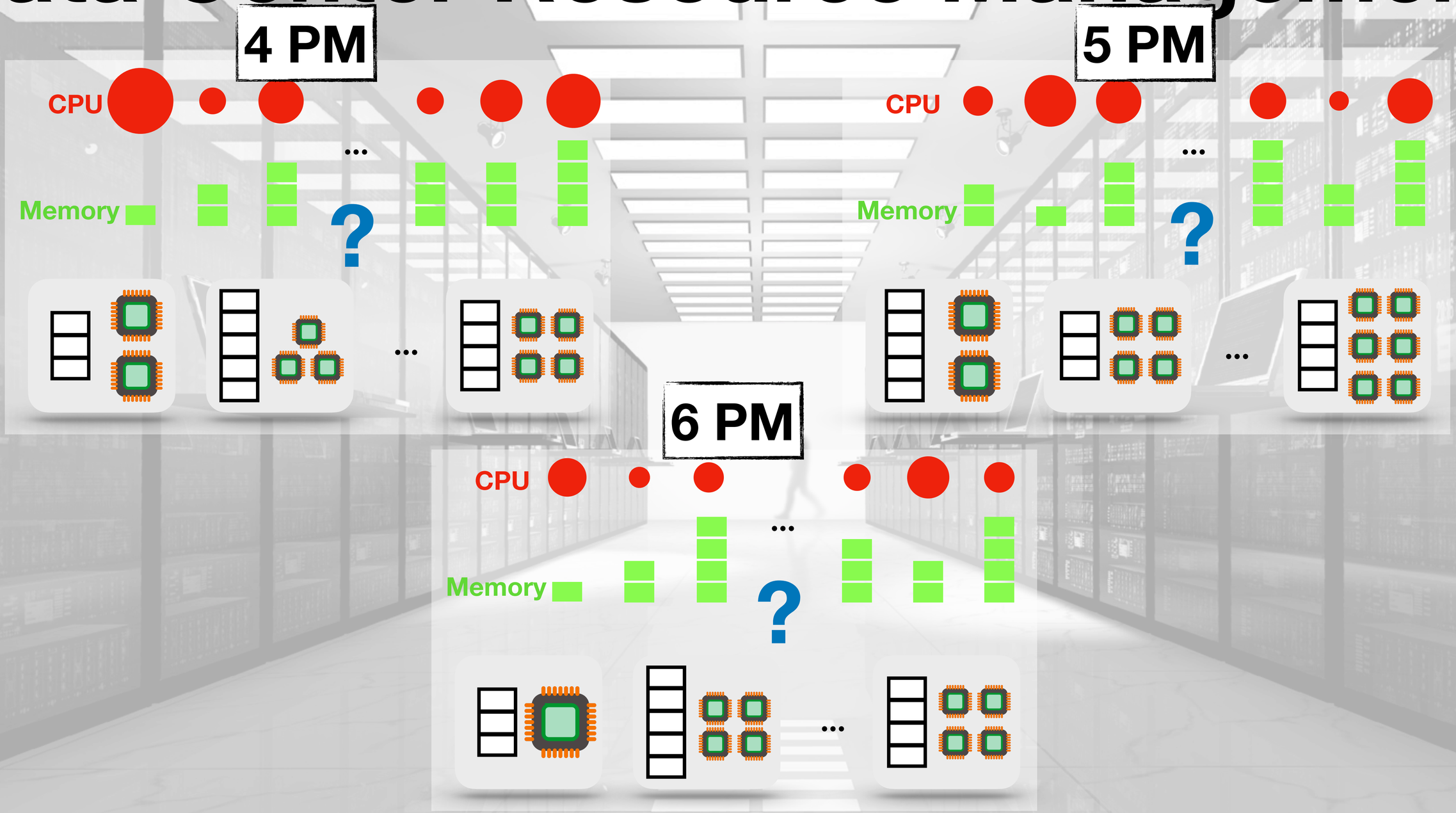
Data Center Resource Management



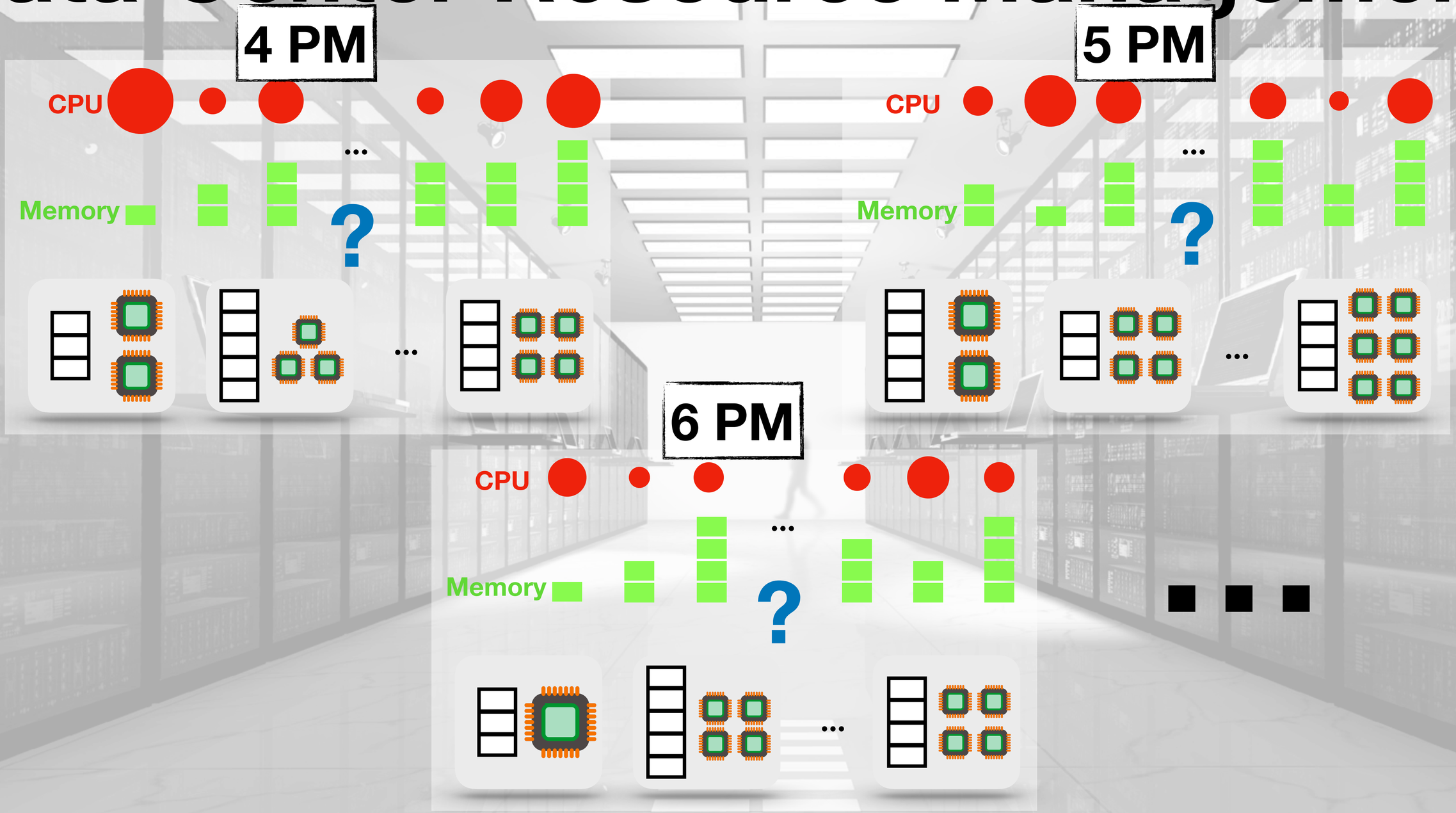
Data Center Resource Management



Data Center Resource Management



Data Center Resource Management



Tackling NP-Hard Problems

Paradigm	Design Rationale

Tackling NP-Hard Problems

Paradigm	Design Rationale
Exhaustive Search	**Tight formulations** Powerful **Branch-and-Bound** solvers

Tackling NP-Hard Problems

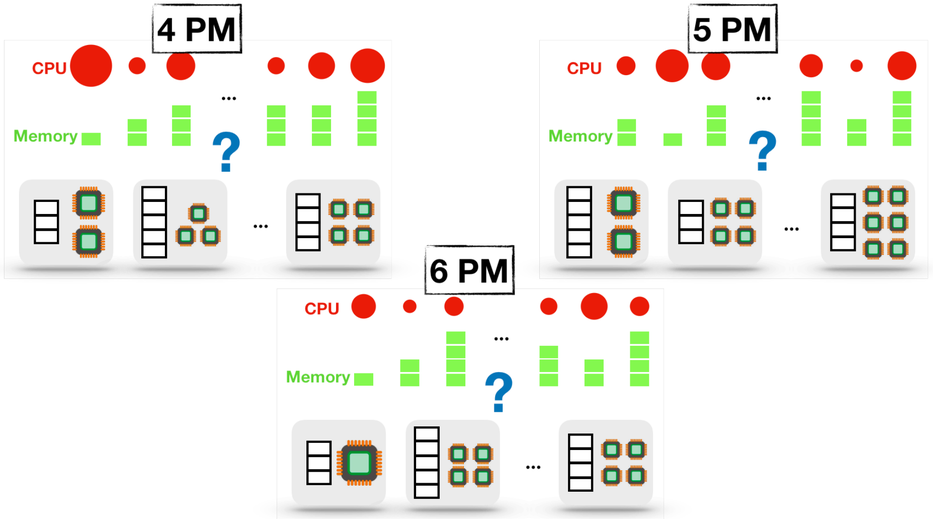
Paradigm	Design Rationale
Exhaustive Search	**Tight formulations** Powerful **Branch-and-Bound** solvers
Approximation Algorithms	**Good worst-case** guarantees

Tackling NP-Hard Problems

Paradigm	Design Rationale
Exhaustive Search	**Tight formulations** Powerful **Branch-and-Bound** solvers
Approximation Algorithms	**Good worst-case** guarantees
Heuristics	**Intuition** exploiting problem structure Empirical **trial-and-error**

Tackling NP-Hard Problems

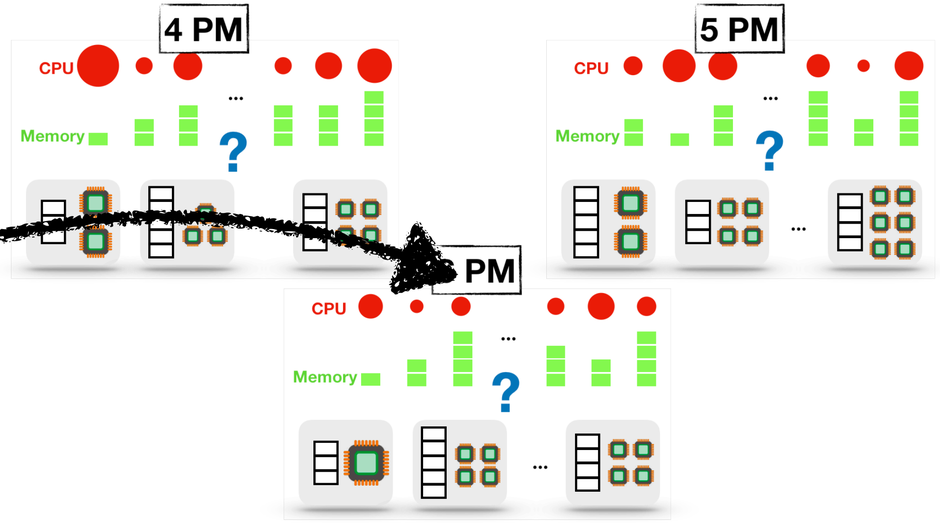
Paradigm	Design Rationale
Exhaustive Search	**Tight formulations** Powerful **Branch-and-Bound** solvers
Approximation Algorithms	**Good worst-case** guarantees
Heuristics	**Intuition** exploiting problem structure Empirical **trial-and-error**



Tackling NP-Hard Problems

Paradigm	Design Rationale
Exhaustive Search	**Tight formulations** Powerful **Branch-and-Bound** solvers
Approximation Algorithms	**Good worst-case** guarantees
Heuristics	**Intuition** exploiting problem structure Empirical **trial-and-error**

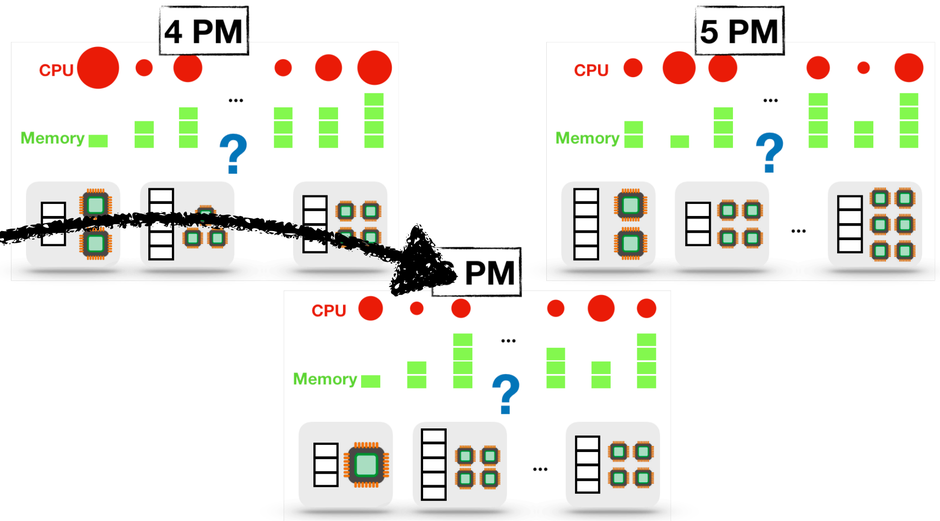
How do you **tailor** the algorithm to **YOUR** instances?



Tackling NP-Hard Problems

Paradigm	Customization via…
Exhaustive Search	**Tight formulations** Powerful **Branch-and-Bound** solvers
Approximation Algorithms	**Good worst-case** guarantees
Heuristics	**Intuition** exploiting problem structure Empirical **trial-and-error**

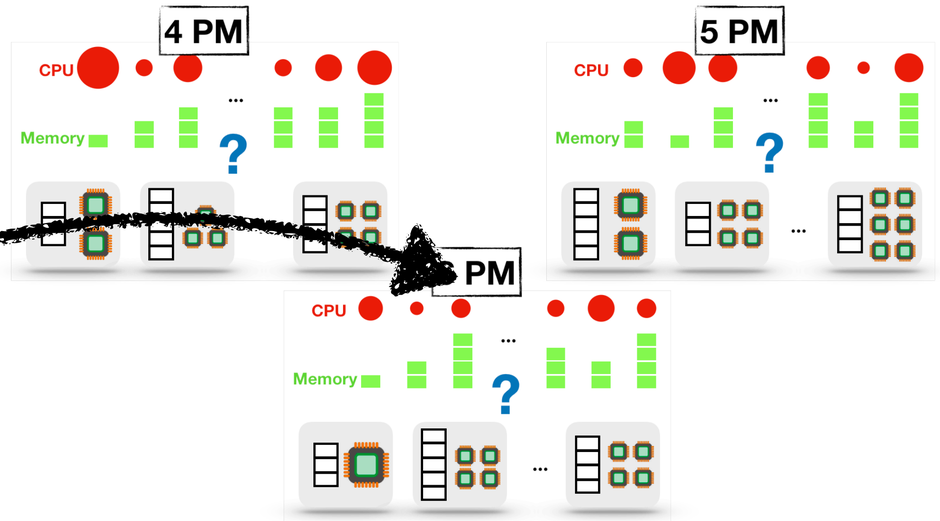
How do you **tailor** the algorithm to **YOUR** instances?



Tackling NP-Hard Problems

Paradigm	Customization via…
Exhaustive Search	**Problem-Specific Bounding** functions or **search rules**
Approximation Algorithms	**Good worst-case** guarantees
Heuristics	**Intuition** exploiting problem structure Empirical **trial-and-error**

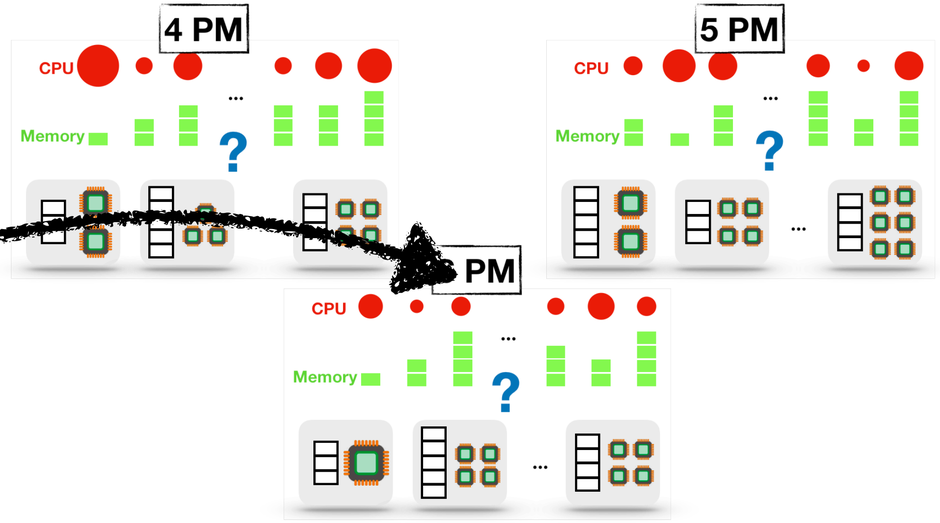
How do you **tailor** the algorithm to **YOUR** instances?



Tackling NP-Hard Problems

Paradigm	Customization via…
Exhaustive Search	**Problem-Specific Bounding** functions or **search rules**
Approximation Algorithms	Make explicit **assumptions** on input distribution and **redesign algo.**
Heuristics	**Intuition** exploiting problem structure Empirical **trial-and-error**

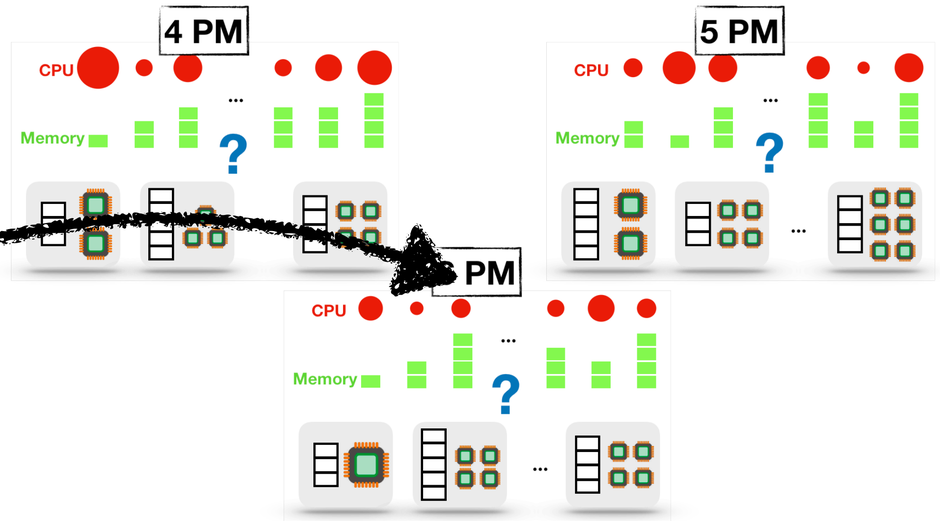
How do you **tailor** the algorithm to **YOUR** instances?



Tackling NP-Hard Problems

Paradigm	Customization via…
Exhaustive Search	**Problem-Specific Bounding** functions or **search rules**
Approximation Algorithms	Make explicit **assumptions** on input distribution and **redesign algo.**
Heuristics	**Analyze** algorithm **behavior** on your inputs; look for **patterns** to exploit

How do you **tailor** the algorithm to **YOUR** instances?

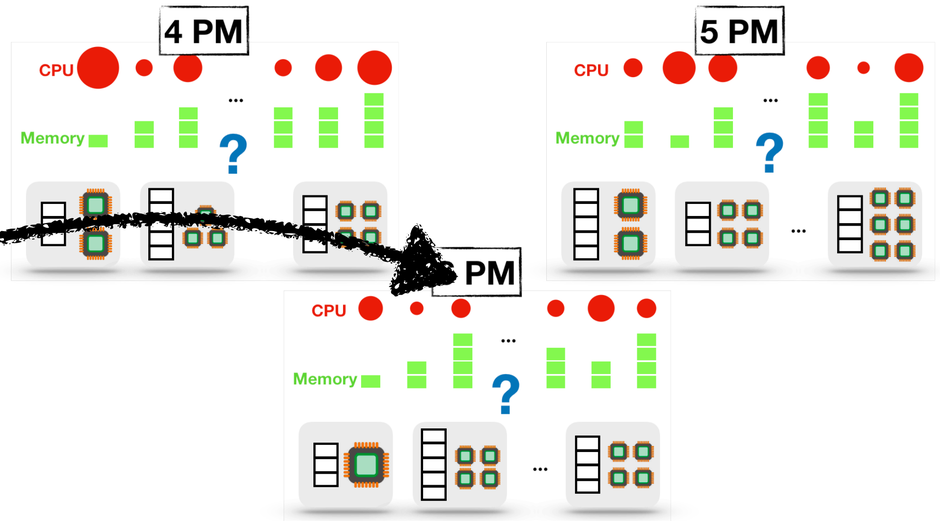


Tackling NP-Hard Problems

Paradigm	Customization via…
	Problem-Specific Bounding functions or **search rules**
	Make explicit **assumptions** on input distribution and **redesign algo.**
	Analyze algorithm **behavior** on your inputs; look for **patterns** to exploit

ANSWER:
Manual intellectual/ experimental **effort** required

How do you **tailor** the algorithm to **YOUR** instances?

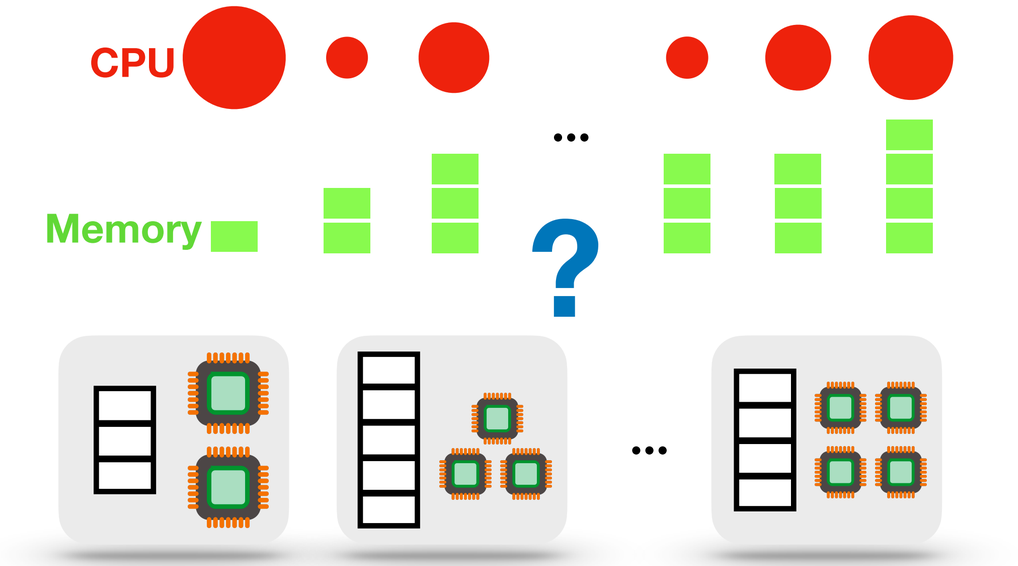


Opportunity

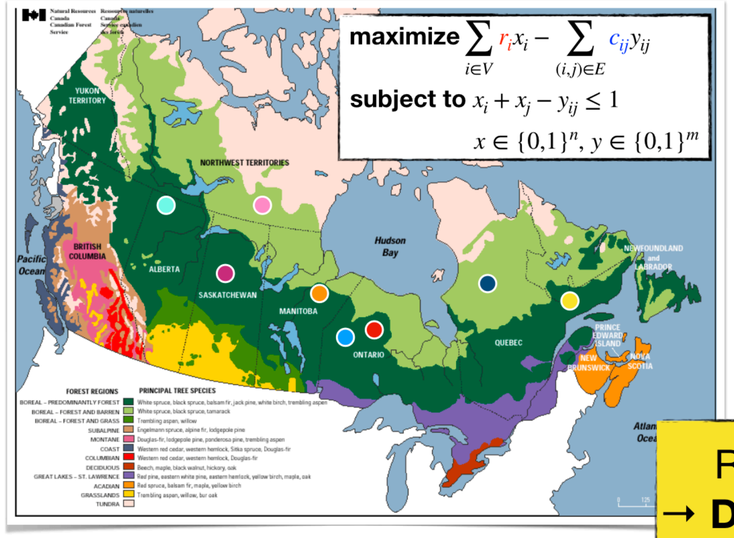
Parts 2-5 of this tutorial

Automatically tailor algorithms to a **family** of **instances**

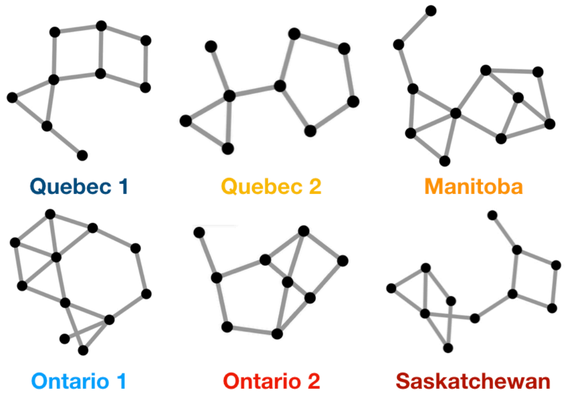
Data Center Resource Management



Forest Harvesting



and many more…



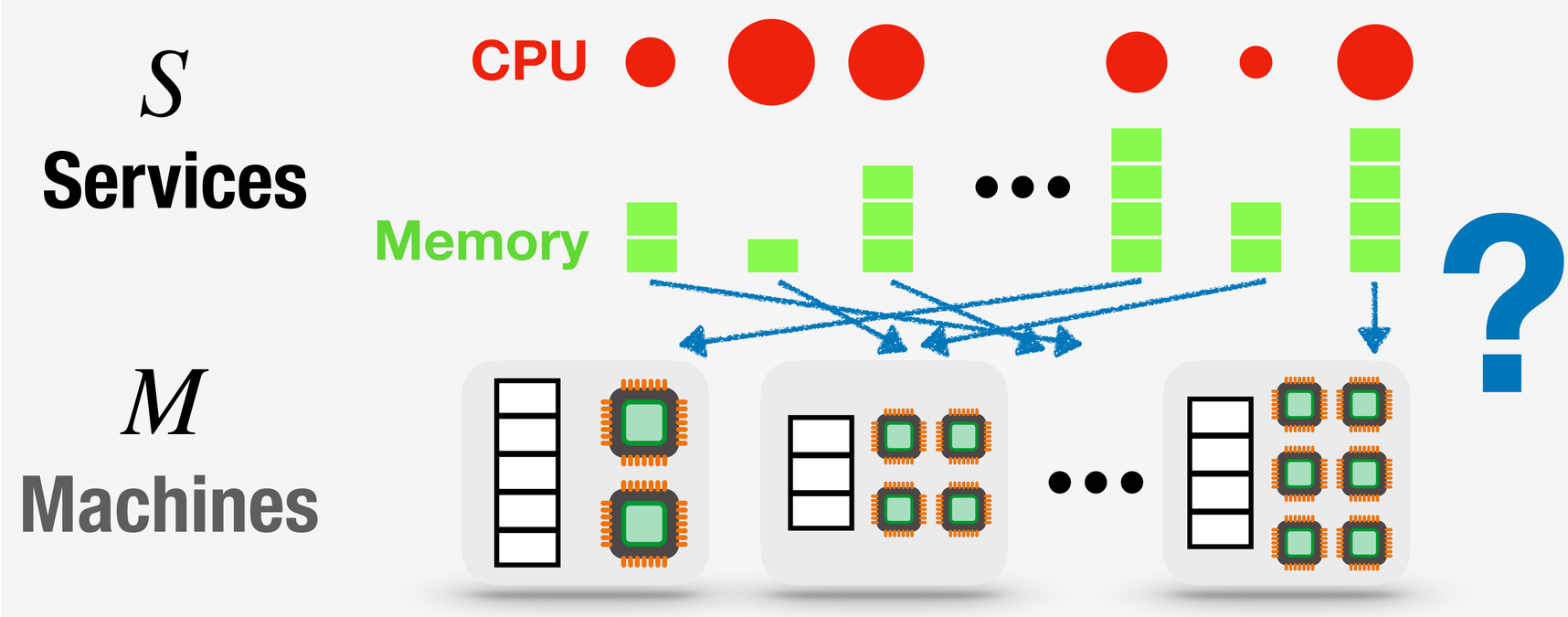
Revenue varies **over time**
→ **Different** instances, **same** problem

$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

$x \in \{0,1\}^{S \times M}, y \in \{0,1\}^M$

$$\textbf{minimize} \sum_{m=1}^{M} y_m$$



Constraints:

Each service on one machine only

$$\sum_{m=1}^{M} x_{s,m} = 1 \quad \forall s$$

$$y_m \geq x_{s,m} \quad \forall s, m$$

Machine is "ON" if a job is assigned to it

Memory capacity

$$\sum_{s=1}^{S} \textbf{mem}(s) \cdot x_{s,m} \leq \textbf{cap-mem}(m) \quad \forall m$$

$$\sum_{s=1}^{S} \textbf{cpu}(s) \cdot x_{s,m} \leq \textbf{cap-cpu}(m) \quad \forall m$$

Processor capacity

$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

$x \in \{0,1\}^{S \times M}, y \in \{0,1\}$

minimize $\sum_{m=1}^{M} y_m$

Where does this information come from?

Constraints:

Each service on one machine only

$$\sum_{m=1}^{M} x_{s,m} = 1 \quad \forall s$$

$$y_m \geq x_{s,m} \quad \forall s, m$$

Machine is "ON" if a job is assigned to it

Memory capacity

$$\sum_{s=1}^{S} \textbf{mem}(s) \cdot x_{s,m} \leq \textbf{cap-mem}(m) \quad \forall m$$

$$\sum_{s=1}^{S} \textbf{cpu}(s) \cdot x_{s,m} \leq \textbf{cap-cpu}(m) \quad \forall m$$

Processor capacity

$y_m = 1$ if machine m is used

$x_{s,m} = 1$ if service s runs on m

$x \in \{0,1\}^{S \times M}, y \in \{0,1\}^M$

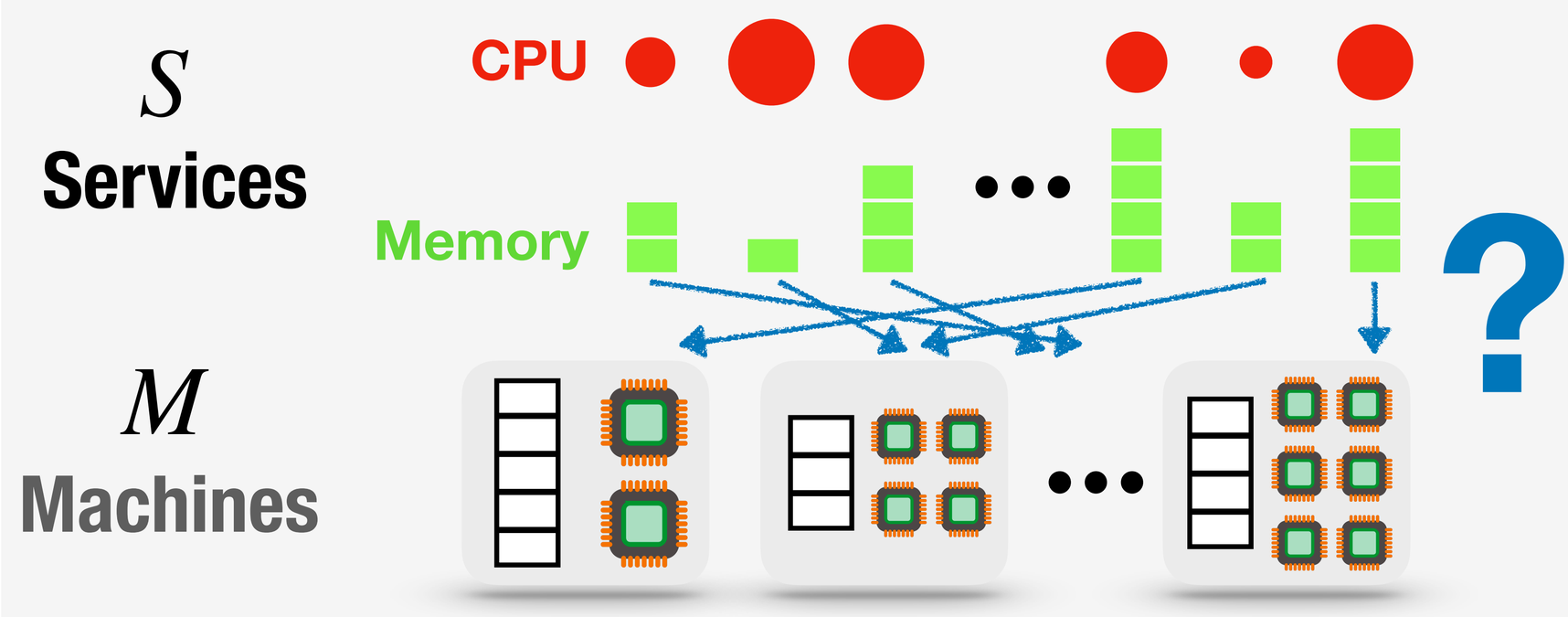
$$\text{minimize} \sum_{m=1}^{M} y_m$$

Constraints:

Each service on one machine only

$$\sum_{m=1}^{M} x_{s,m} = 1 \quad \forall s$$

Machine is "ON" if a job is assigned to it



Memory capacity

$$\sum_{s=1}^{S} \textbf{mem}(s) \cdot x_{s,m} \leq \textbf{cap-mem}(m) \quad \forall m$$

$$\sum_{s=1}^{S} \textbf{cpu}(s) \cdot x_{s,m} \leq \textbf{cap-cpu}(m) \quad \forall m$$

Processor capacity

ML model for CPU usage:

$\textbf{cpu}(s) = \textbf{model}_\theta(\textbf{features}(s))$

Opportunity

Part 6 of this tutorial

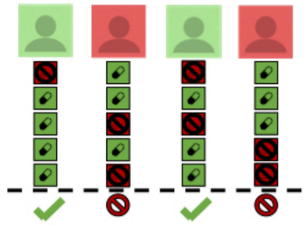
Decision-Focused Learning:

ML models that accurately predict parameters of optimization problems

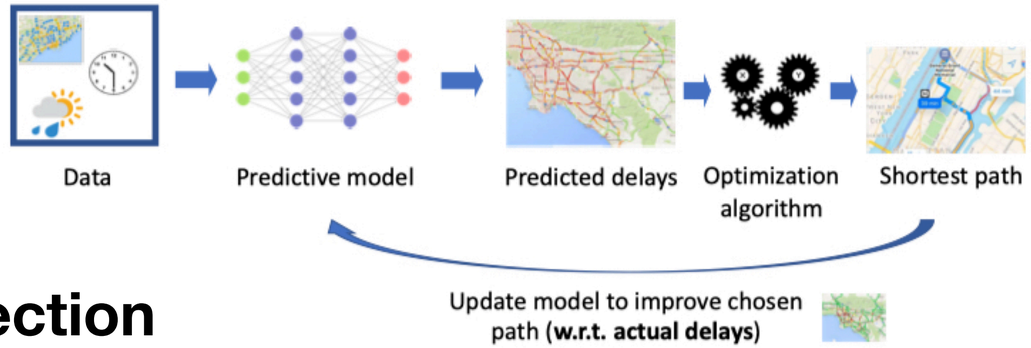
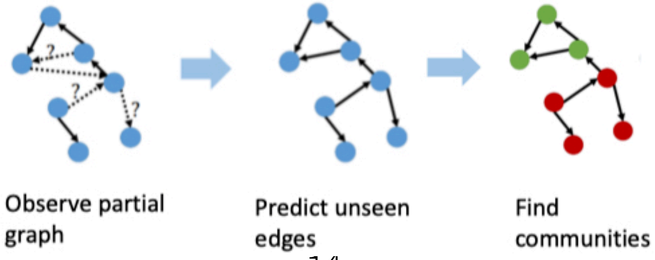
Tuberculosis Treatment

- **Predict** if patients will miss daily dose
- **Optimize** health worker visits subject to knapsack constraints (LP)

- More in our paper [Killian et al, KDD 2019]



Shortest paths with predictions



Community Detection